Introducción a la Criptografía Moderna

Algoritmo Simétrico de Encriptación

Advanced Encryption Standard, AES

Rodrigo Abarzúa[†],

[†] Universidad de Santiago de Chile
rodrigo.abarzua@usach.cl

April 17, 2014

Advanced Encryption Standard, AES

- En 1997 NIST llamo a concurso para proponer el nuevo Advanced Encryption Standard (AES).
- La selección del algoritmo para AES fue un proceso público administrado por NIST.
- En tres rondas de evaluación NIST y la comunidad científica internacional discutió las ventajas y desventajas de los cifradores sometidos en el concurso.
- En el 2001, NIST declaro que el cifrador de bloques *Rijndael* es el nuevo AES y publicó el estándar final FIPS PUB 197.
- Rijndael fue diseñado por los criptógrafos Belgas Joan Daemen y Vincent Rijmen ambos estudiantes de la *Katholieke Universiteit Leuven*.

Advanced Encryption Standard, AES

Los requisitos que debía cumplir este nuevo criptosistema eran:

- Cifrado de bloque, con bloques de 128-bits.
- Debía soportar claves de longitud: 128, 192 y 256 bit.
- Seguridad en relación a los otros criptosistemas presentados en el concurso.
- Eficiente en software como en hardware.

Advanced Encryption Standard, AES

Los requisitos que debía cumplir este nuevo criptosistema eran:

- Cifrado de bloque, con bloques de 128-bits.
- Debía soportar claves de longitud: 128, 192 y 256 bit.
- Seguridad en relación a los otros criptosistemas presentados en el concurso.
- Eficiente en software como en hardware.

Advanced Encryption Standard, AES

Los requisitos que debía cumplir este nuevo criptosistema eran:

- Cifrado de bloque, con bloques de 128-bits.
- Debía soportar claves de longitud: 128, 192 y 256 bit.
- Seguridad en relación a los otros criptosistemas presentados en el concurso.
- Eficiente en software como en hardware.

Advanced Encryption Standard, AES

Los requisitos que debía cumplir este nuevo criptosistema eran:

- Cifrado de bloque, con bloques de 128-bits.
- Debía soportar claves de longitud: 128, 192 y 256 bit.
- Seguridad en relación a los otros criptosistemas presentados en el concurso.
- Eficiente en software como en hardware.

Advanced Encryption Standard, AES

El *"Advanced Encryption Standard (AES)"* es el criptosistema simétrico más ampliamente usado.

- El AES es un cifrador obligatorio en varios estándares y utilizado en muchos sistemas comerciales.
- Entre los estándares comerciales que incluye el AES son los estándares de seguridad de Internet como el IPsec, TLS, el WI-FI encryption standard IEEE 802.11i, ect.

Advanced Encryption Standard, AES

En el 2003 el US National Security Agency (NSA, http://www.nsa.gov/) anuncio que se permite el AES para encriptar documentos clasificados como nivel:

- SECRET para todas las longitudes de claves (128, 192 y 256 bit).
- TOP SECRET para claves de longitud 192 y 256 bits.

Advanced Encryption Standard, AES

El AES es un cifrador de bloque de longitud de 128 bits.

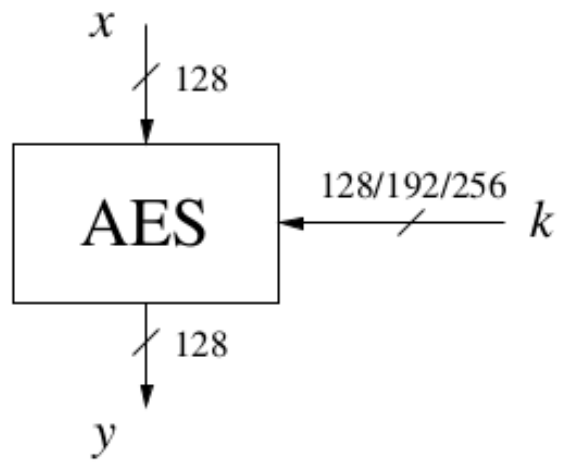


Figure: Parametros de entrada y salida del AES

Advanced Encryption Standard, AES

Dependiendo del tamaño de la clave varía el número de rondas de ejecución del algoritmo, en la siguiente tabla se presentan la relación longitud de la clave y el número de iteraciones correspondiente.

Key length	Número de rondas (n_r)
128 bit	10
192 bit	12
256 bit	14

Table: Longitud de la clave y el número de rondas para AES

Advanced Encryption Standard, AES

- Al contrario del DES, el AES no tiene una estructura de Feistel. La red Feistel no encripta el bloque completo por iteración
- El AES encripta todos los 128 bits en una iteración. Esto en una de las razones porque el AES tiene menor número de rondas.
- AES consiste en lo que se llama capaz *"layers"*. En cada capa son manipuladas todos los 128-bits del paquete de información.
- Los paquetes de data son también conocidos como los estados del algoritmo. Sólo hay tres diferentes tipos capaz.
- En cada ronda, excepto en la primera, consiste en tres capaz como se muestra en la siguiente figura.
- El plaintext es denotado por x y el ciphertext como y el número de rondas es n_r.
- Además, la última ronda n_r no utiliza la transformación **MixColumn**, que hace que encriptar y la desencriptar tengan un esquema simétrico.

Advanced Encryption Standard, AES

- Al contrario del DES, el AES no tiene una estructura de Feistel. La red Feistel no encripta el bloque completo por iteración
- El AES encripta todos los 128 bits en una iteración. Esto en una de las razones porque el AES tiene menor número de rondas.
- AES consiste en lo que se llama capaz *"layers"*. En cada capa son manipuladas todos los 128-bits del paquete de información.
- Los paquetes de data son también conocidos como los estados del algoritmo. Sólo hay tres diferentes tipos capaz.
- En cada ronda, excepto en la primera, consiste en tres capaz como se muestra en la siguiente figura.
- El plaintext es denotado por x y el ciphertext como y el número de rondas es n_r.
- Además, la última ronda n_r no utiliza la transformación **MixColumn**, que hace que encriptar y la desencriptar tengan un esquema simétrico.

Advanced Encryption Standard, AES

- Al contrario del DES, el AES no tiene una estructura de Feistel. La red Feistel no encripta el bloque completo por iteración
- El AES encripta todos los 128 bits en una iteración. Esto en una de las razones porque el AES tiene menor número de rondas.
- AES consiste en lo que se llama capaz *"layers"*. En cada capa son manipuladas todos los 128-bits del paquete de información.
- Los paquetes de data son también conocidos como los estados del algoritmo. Sólo hay tres diferentes tipos capaz.
- En cada ronda, excepto en la primera, consiste en tres capaz como se muestra en la siguiente figura.
- El plaintext es denotado por x y el ciphertext como y el número de rondas es n_r.
- Además, la última ronda n_r no utiliza la transformación **MixColumn**, que hace que encriptar y la desencriptar tengan un esquema simétrico.

Advanced Encryption Standard, AES

- Al contrario del DES, el AES no tiene una estructura de Feistel. La red Feistel no encripta el bloque completo por iteración
- El AES encripta todos los 128 bits en una iteración. Esto en una de las razones porque el AES tiene menor número de rondas.
- AES consiste en lo que se llama capaz *"layers"*. En cada capa son manipuladas todos los 128-bits del paquete de información.
- Los paquetes de data son también conocidos como los estados del algoritmo. Sólo hay tres diferentes tipos capaz.
- En cada ronda, excepto en la primera, consiste en tres capaz como se muestra en la siguiente figura.
- El plaintext es denotado por x y el ciphertext como y el número de rondas es n_r.
- Además, la última ronda n_r no utiliza la transformación **MixColumn**, que hace que encriptar y la desencriptar tengan un esquema simétrico.

Advanced Encryption Standard, AES

- Al contrario del DES, el AES no tiene una estructura de Feistel. La red Feistel no encripta el bloque completo por iteración
- El AES encripta todos los 128 bits en una iteración. Esto en una de las razones porque el AES tiene menor número de rondas.
- AES consiste en lo que se llama capaz *"layers"*. En cada capa son manipuladas todos los 128-bits del paquete de información.
- Los paquetes de data son también conocidos como los estados del algoritmo. Sólo hay tres diferentes tipos capaz.
- En cada ronda, excepto en la primera, consiste en tres capaz como se muestra en la siguiente figura.
- El plaintext es denotado por x y el ciphertext como y el número de rondas es n_r.
- Además, la última ronda n_r no utiliza la transformación **MixColumn**, que hace que encriptar y la desencriptar tengan un esquema simétrico.

Advanced Encryption Standard, AES

- Al contrario del DES, el AES no tiene una estructura de Feistel. La red Feistel no encripta el bloque completo por iteración
- El AES encripta todos los 128 bits en una iteración. Esto en una de las razones porque el AES tiene menor número de rondas.
- AES consiste en lo que se llama capaz *"layers"*. En cada capa son manipuladas todos los 128-bits del paquete de información.
- Los paquetes de data son también conocidos como los estados del algoritmo. Sólo hay tres diferentes tipos capaz.
- En cada ronda, excepto en la primera, consiste en tres capaz como se muestra en la siguiente figura.
- El plaintext es denotado por x y el ciphertext como y el número de rondas es n_r.
- Además, la última ronda n_r no utiliza la transformación **MixColumn**, que hace que encriptar y la desencriptar tengan un esquema simétrico.

Advanced Encryption Standard, AES

- Al contrario del DES, el AES no tiene una estructura de Feistel. La red Feistel no encripta el bloque completo por iteración
- El AES encripta todos los 128 bits en una iteración. Esto en una de las razones porque el AES tiene menor número de rondas.
- AES consiste en lo que se llama capaz *"layers"*. En cada capa son manipuladas todos los 128-bits del paquete de información.
- Los paquetes de data son también conocidos como los estados del algoritmo. Sólo hay tres diferentes tipos capaz.
- En cada ronda, excepto en la primera, consiste en tres capaz como se muestra en la siguiente figura.
- El plaintext es denotado por x y el ciphertext como y el número de rondas es n_r.
- Además, la última ronda n_r no utiliza la transformación **MixColumn**, que hace que encriptar y la desencriptar tengan un esquema simétrico.

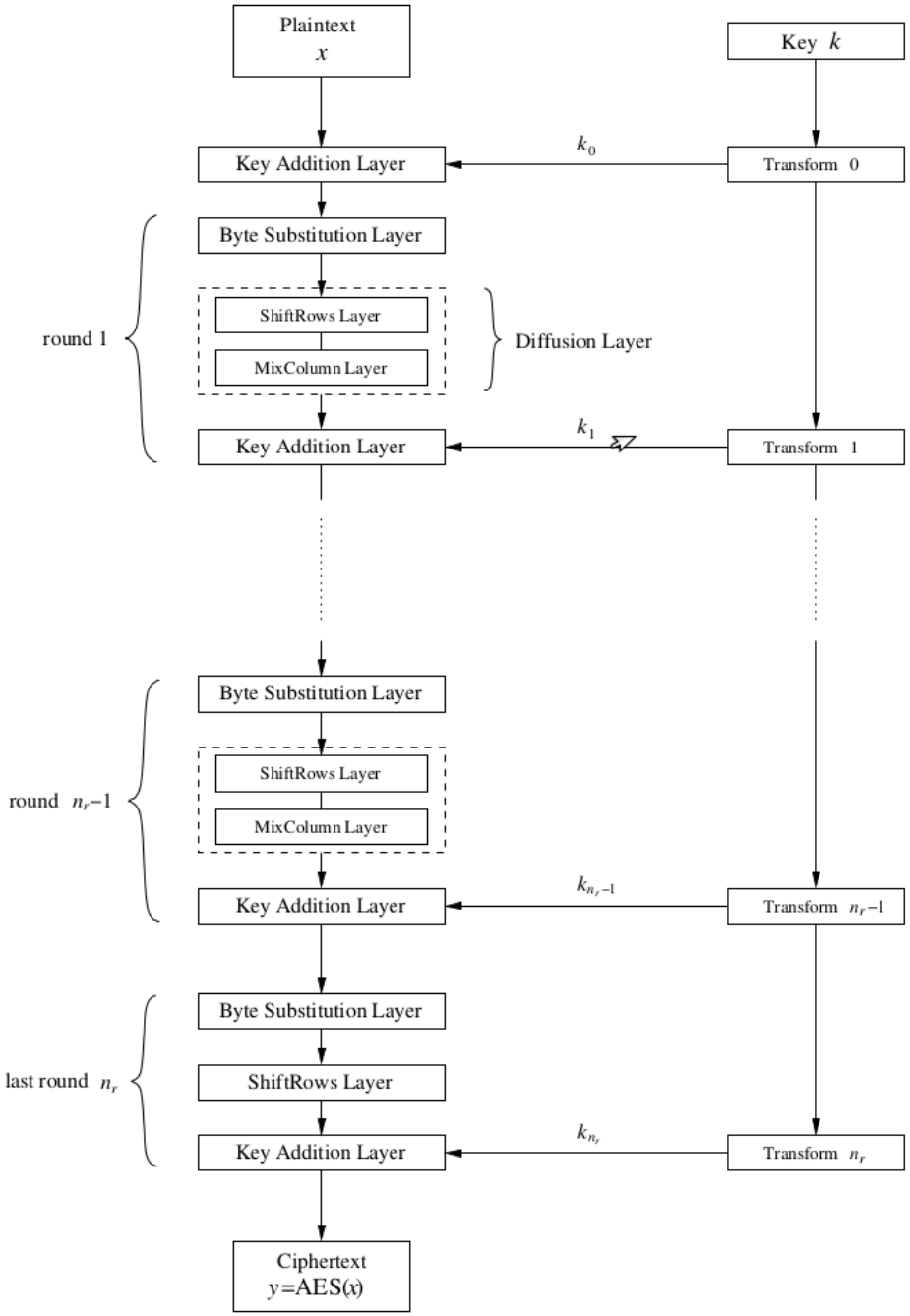


Figure: Diagrama de bloque del encriptación del AES

Advanced Encryption Standard, AES

Describamos cada capa del AES:

Key Addition Layer : Es una ronda de la clave de 128-bits, o subclave que se derivan de una clave principal en el *key shedule*, es un XORed del estado.

Substitución de Byte (S-Box) Layer : Cada elemento del estado es una transformaci.ón no-lineal usando una tabla de búsqueda con propiedades matemáticas especiales. Esto introduce *confusión* de la data, es decir, asegura que cambios en estados individuales de los bits se propagan rápidamente a través del proceso iterativo del AES.

Definición

La confusión es una operación de cifrado es donde la relación entre la clave y texto cifrado es oscurecida. Hoy en día, un elemento común para el logro de confusión es la sustitución, que se encuentra tanto en DES y AES.

Advanced Encryption Standard, AES

Describamos cada capa del AES:

Layer de Difusión: Esta capa provee difusión sobre todos los bits de estados. Esto consiste en dos subcapas, ambas desempeñan operaciones lineales:

- La capa *ShiftRows* es una permutación de la información a nivel de bits.
- La capa *MixColumn* es una matriz operacional que combina o mezcla bloques de 4 bytes.

Key Schedule Layer: Al igual que el DES, la *Key Schedule* calcula la ronda de claves o subclaves, $(k_0, k_1, \ldots, k_{n_r})$ de la clave original del AES.

Definición

La difusión es una operación de cifrado, donde la influencia de un símbolo de texto claro se extiende sobre muchos símbolos de texto cifrado con el objetivo de ocultar propiedades estadísticas del plaintext. Un elemento de difusión simple es la permutación de bit, que es utilizado con frecuencia dentro de DES. En el AES utiliza la operación más avanzada MixColumn.

Observación

Para el AES:

- *El polinomio irreducible que se utiliza es el*

$$P(x) = x^8 + x^4 + x^3 + x + 1$$

- *Para el cuerpo utilizado en el AES se utilizan los lookup tables que contiene precalculos de todos los inversos del cuerpo. Un caso especiales es para la entrada del elemento del cuerpo* 0 *el cual el inverso no existe.*

Observación

- *Sin embargo para el AES S-Box, la tabla de sustitución debe ser definida para todos los valores de entrada. Luego, el diseño para definir las S-Box se construye para que el valor* 0 *es mapeado como salida* 0

	Y															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	8D	F6	CB	52	7B	D1	E8	4F	29	C0	B0	E1	E5	C7
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
5	ED	5C	05	CA	4C	24	87	BF	18	3E	22	F0	51	EC	61	17
6	16	5E	AF	D3	49	A6	36	43	F4	47	91	DF	33	93	21	3B
7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	A1	FA	81	82
X 8	83	7E	7F	80	96	73	BE	56	9B	9E	95	D9	F7	02	B9	A4
9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A
A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62
B	0C	E0	1F	EF	11	75	78	71	A5	8E	76	3D	BD	BC	86	57
C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6
D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B
E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3
F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	1A	41	1C

Observación

Entonces por ejemplo de la tabla anterior el inverso de

$$x^7 + x^6 + x = (11000010)_2 = (C2)_{hex} = (xy)$$

es dado por el elemento de la fila de C y la columna 2:

$$(2F)_{hex} = (00101111)_2 = x^5 + x^3 + x^2 + x + 1.$$

Se puede verificar que:

$$(x^7 + x^6 + x) \cdot (x^5 + x^3 + x^2 + x + 1) \equiv 1 \bmod P(x)$$

Otra alternativa para calcular el inverso es utilizar el algoritmo Euclidiano extendido

Advanced Encryption Standard, AES

El cambio desde el DES al AES se debe fundamentalmente que:

- DES no es eficiente en software.
- DES sólo opera con bloques de tamaño de 64 bits, que es un inconveniente cuando por ejemplo se desea construir una función de hash desde un cifrado de bloques (lo estudiaremos más adelante).
- Otro problema que se ve venir es la tecnología de computadores cuánticos se desea longitudes de clave de orden de 256 bits.

Advanced Encryption Standard, AES

En la siguiente figura presentamos la estructura interna del AES. Se presenta una sola ronda del AES.

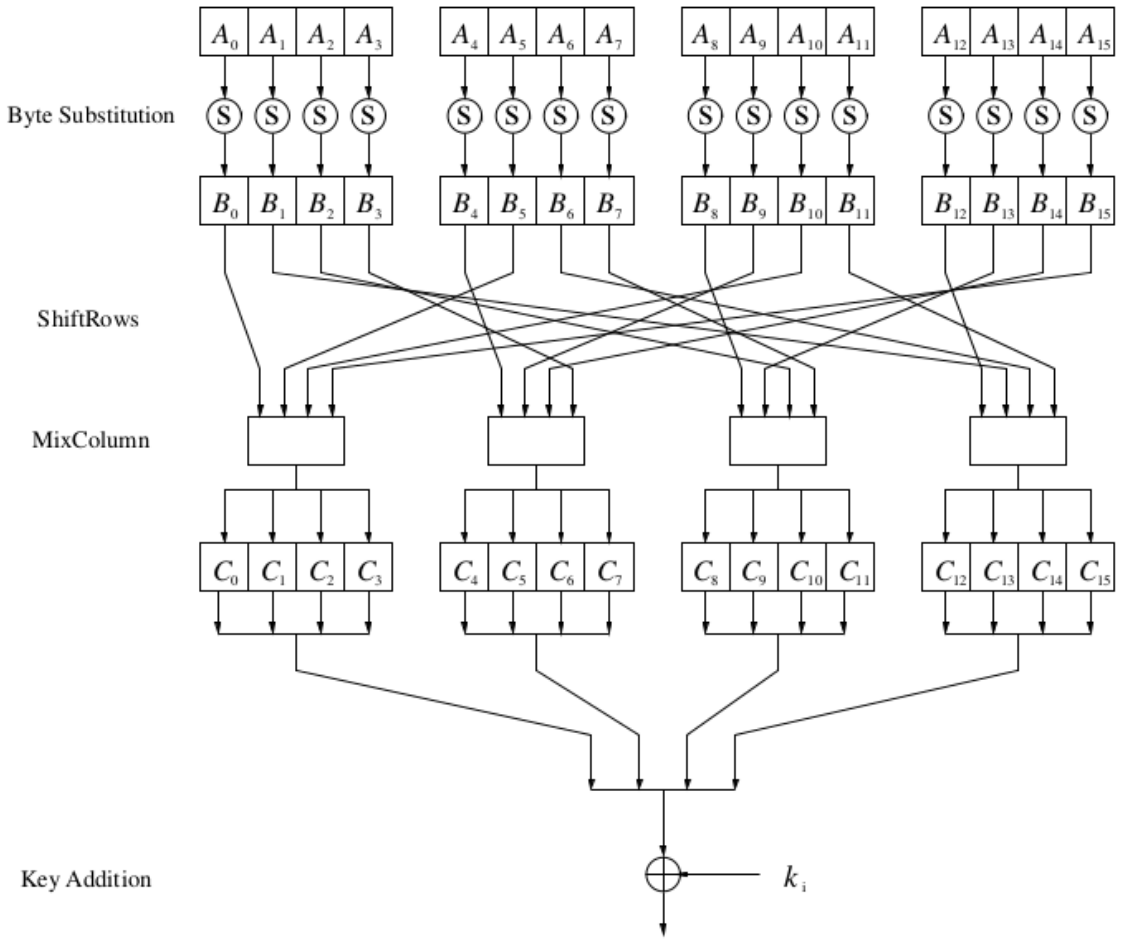


Figure: AES, para las rondas $1, 2, \ldots, n_r - 1$

Advanced Encryption Standard, AES

- Los 16-bytes de entradas $A_0, \ldots, A_{15}$ son ingresados por bytes en los S-Box.
- El 16−byte de salida $B_0, \ldots, B_{15}$ son permutados byte a byte en la capa o función **ShiftRows** y mezclado con la transformación **MixColumn** $C(x)$.
- Finalmente, las subclaves k_i de 128-bit son *XORed* con el resultado interno. Notemos que el AES, es un cifrador orientado a los byte.

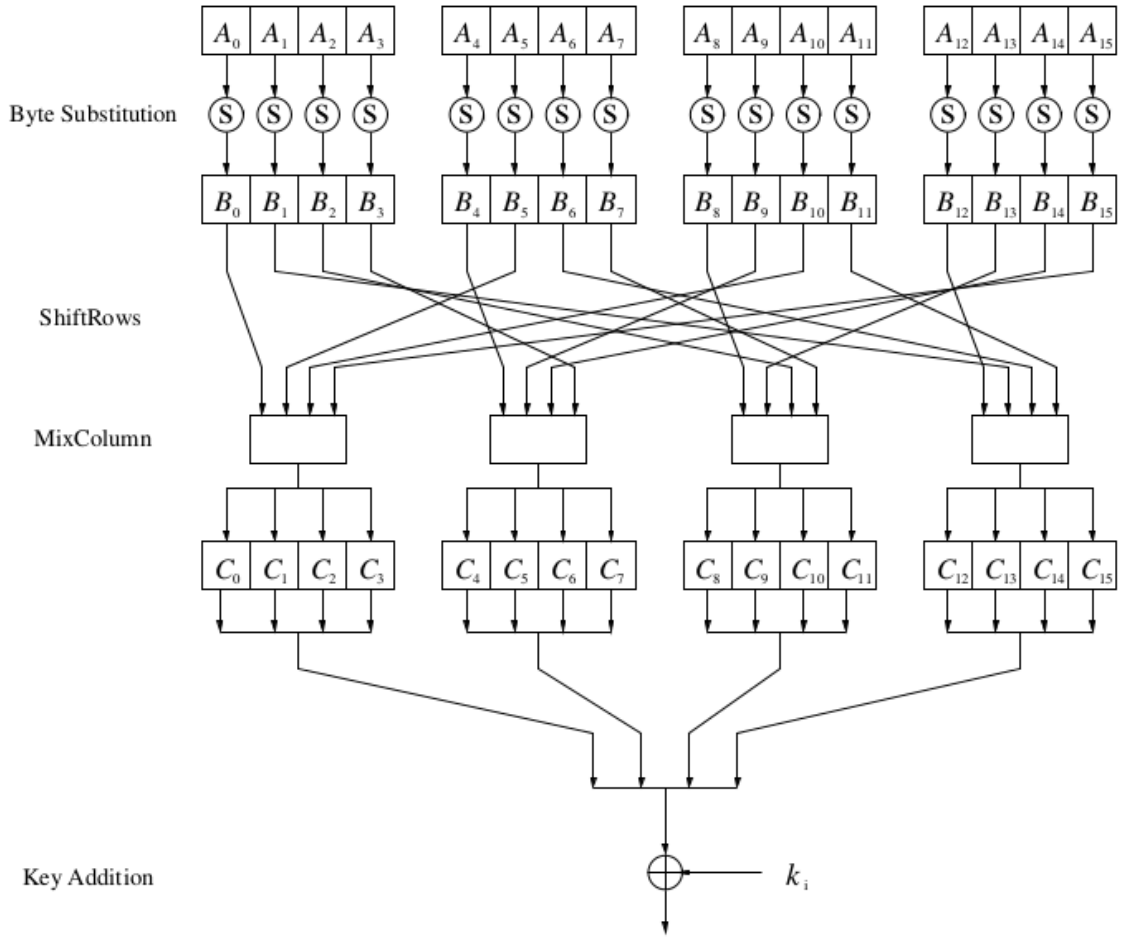


Figure: AES, para las rondas $1, 2, \ldots, n_r - 1$

Advanced Encryption Standard, AES

- Al contrario que el DES, el cual manipula la información a nivel de "bit" por lo tanto se puede considerar que el DES tiene una estructura orientada al bit.
- Por otro lado, ell AES manipula bytes, luego se considera al AES tiene una estructura orientada a bytes.

Advanced Encryption Standard, AES

- Para entender como el AES mueve la información, primero tomemos un paquete de información A de 128-bit consistente de 16 bytes: $A_0, A_1, \ldots, A_{15}$ en ordenado en una matriz de 4 por 4, como la siguiente figura:

A_0	A_4	A_8	A_{12}
A_1	A_5	A_9	A_{13}
A_2	A_6	A_{10}	A_{14}
A_3	A_7	A_{11}	A_{15}

Para la Clave k

- El AES opera los elementos por columnas o filas de la matriz de estado actual.
- Análogamente, los bytes de la clave están ordenados en una matriz de 4 por:
  - 4 columnas de claves de 128-bit .
  - 6 columnas de claves de 192-bit .
  - 8 columnas de claves de 256-bit .

Por ejemplo, para la matriz de estado para una clave de 192-bit.

k_0	k_4	k_8	k_{12}	k_{16}	k_{20}
k_1	k_5	k_9	k_{13}	k_{17}	k_{21}
k_2	k_6	k_{10}	k_{14}	k_{18}	k_{22}
k_3	k_7	k_{11}	k_{15}	k_{19}	k_{23}

Advanced Encryption Standard, AES

En los sigue estudiaremos cada capa del AES.

1. Byte Substitution Layer
2. Diffusion Layer
3. Key Addition Layer
4. Key Schedule

Advanced Encryption Standard, AES

- La primera capa de cada ronda es el *Byte Substitution Layer*.
- Esta capa se puede ver como 16 filas en paralelo aplicadas en S-Boxes cada una con entradas y salidas de 8 bits.
- Se debe notar que todos los 16 S-Boxes son idénticos a diferencia de DES cuando se utilizan ocho diferentes S-Boxes.

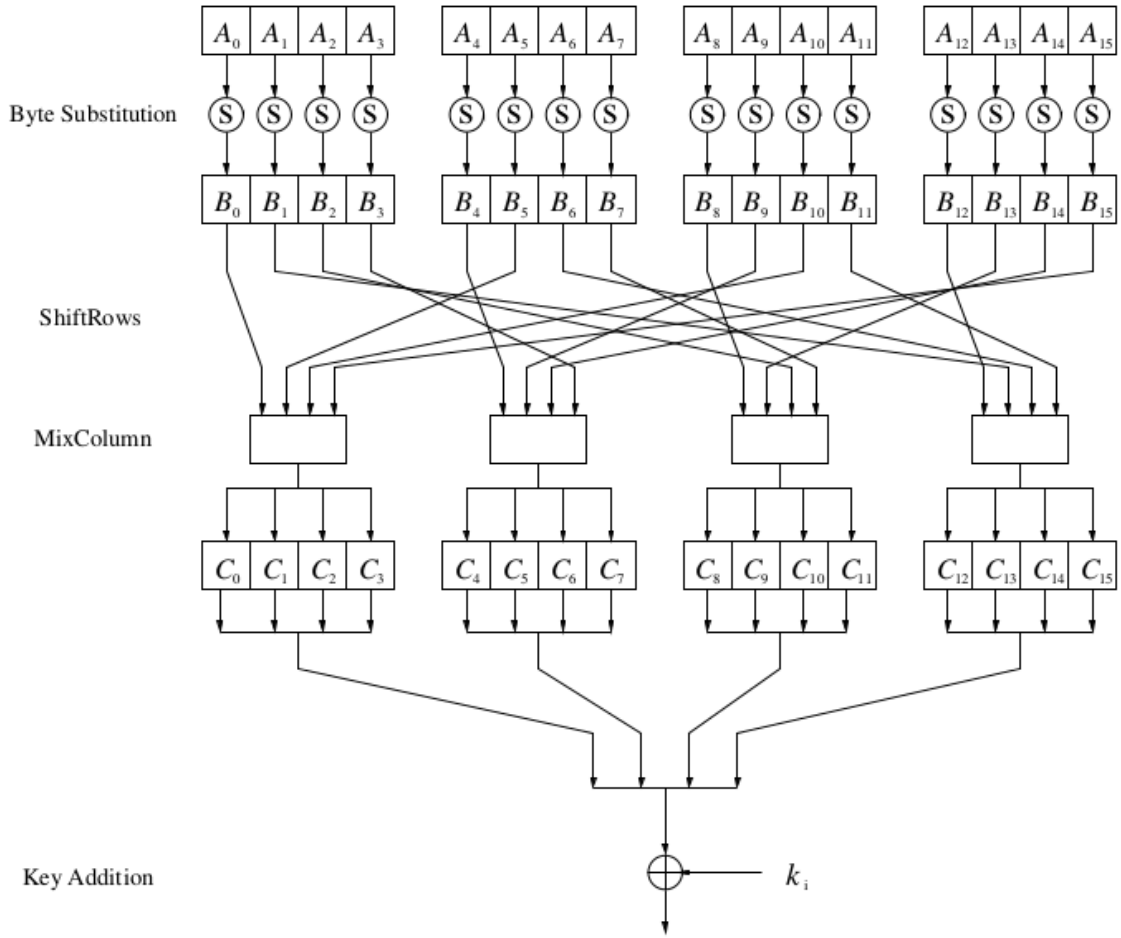


Figure: AES, para las rondas $1, 2, \ldots, n_r - 1$

Advanced Encryption Standard, AES

- En esta capa, cada estado de byte A_i es sustituido por otro byte B_i:

$$S(A_i) = B_i.$$

- La descripción de la función S es puede ver a través de dos transformaciones matemáticas como en la siguiente figura:

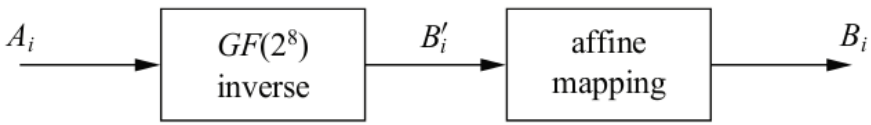


Figure: Las dos operaciones que se pueden ver en el S-Box para $S(A_i) = B_i$

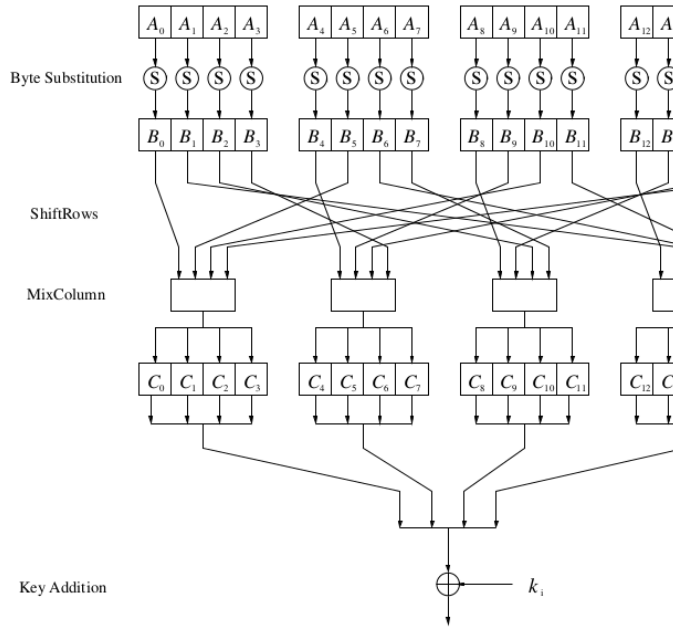


Figure: AES, para las rondas $1, 2, \ldots$

Advanced Encryption Standard, AES

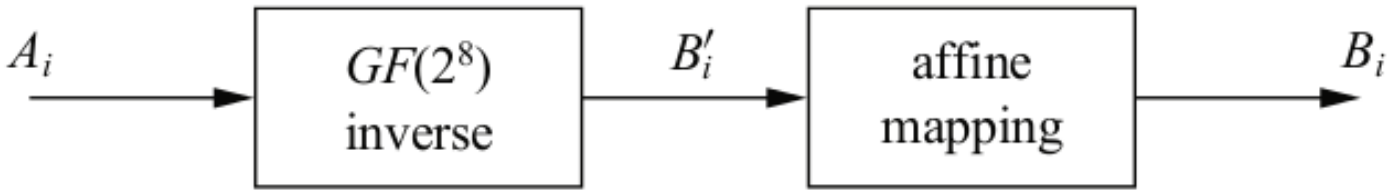


Figure: Las dos operaciones que se pueden ver en el S-Box para $S(A_i) = B_i$

- La primera parte de la sustitución es una inversa en el cuerpo de Galois $GF(2^8)$.
- Es decir, para cada entrada A_i se calcula

$$B_i^{'} = A_i^{-1}$$

  donde A_i y $B_i^{'}$ se considera como elementos de $GF(2^8)$.
- Y en el polinomio irreducible $P(x) = x^8 + x^4 + x^3 + x + 1$, que caracteriza a $GF(2^8)$.

Advanced Encryption Standard, AES

- La lookup tabla de todas los inversos se puede ver en la siguiente figura.
- Se debe observar que el elemento cero no tiene inverso. Sin embargo, para el AES se define el elemento $A_i = 0$ es llevado a si mismo.

	Y															
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	00	01	8D	F6	CB	52	7B	D1	E8	4F	29	C0	B0	E1	E5	C7
1	74	B4	AA	4B	99	2B	60	5F	58	3F	FD	CC	FF	40	EE	B2
2	3A	6E	5A	F1	55	4D	A8	C9	C1	0A	98	15	30	44	A2	C2
3	2C	45	92	6C	F3	39	66	42	F2	35	20	6F	77	BB	59	19
4	1D	FE	37	67	2D	31	F5	69	A7	64	AB	13	54	25	E9	09
5	ED	5C	05	CA	4C	24	87	BF	18	3E	22	F0	51	EC	61	17
6	16	5E	AF	D3	49	A6	36	43	F4	47	91	DF	33	93	21	3B
7	79	B7	97	85	10	B5	BA	3C	B6	70	D0	06	A1	FA	81	82
X 8	83	7E	7F	80	96	73	BE	56	9B	9E	95	D9	F7	02	B9	A4
9	DE	6A	32	6D	D8	8A	84	72	2A	14	9F	88	F9	DC	89	9A
A	FB	7C	2E	C3	8F	B8	65	48	26	C8	12	4A	CE	E7	D2	62
B	0C	E0	1F	EF	11	75	78	71	A5	8E	76	3D	BD	BC	86	57
C	0B	28	2F	A3	DA	D4	E4	0F	A9	27	53	04	1B	FC	AC	E6
D	7A	07	AE	63	C5	DB	E2	EA	94	8B	C4	D5	9D	F8	90	6B
E	B1	0D	D6	EB	C6	0E	CF	AD	08	4E	D7	E3	5D	50	1E	B3
F	5B	23	38	34	68	46	03	8C	DD	9C	7D	A0	CD	1A	41	1C

Advanced Encryption Standard, AES

En la segunda parte de la sustitución, cada byte $B_i^{'}$ es multiplicado por una matriz contante de bit seguida de una adición por un vector constante de $8-$bit. La operación se describe por:

$$\begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} \equiv \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} b_0^{'} \\ b_1^{'} \\ b_2^{'} \\ b_3^{'} \\ b_4^{'} \\ b_5^{'} \\ b_6^{'} \\ b_7^{'} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix} \mod 2$$

Este segundo paso se denomina *"affine mapping"*.

Advanced Encryption Standard, AES

- El S-box es un elemento no lineal del AES, es decir, dados dos estamos A y B

$$ByteSub(A + B) \neq ByteSub(A) + ByteSub(B)$$

- Además, la sustitución S-Box es una función biyectiva, para los $2^8 = 256$ posibles entradas. Esto permite tener una única S-Box inversa necesaria para desencriptar.

Ejemplo

Supongamos que una entrada $A_i = (11000010)_2 = (C2)_{hex}$, *entonces al observar la Figura 3 podemos calcular el inverso:*

$$A_i^{-1} = B_i^{'} = (2F)_{hex} = (00101111)_2.$$

Aplicando la transformación afín al vector $B_i^{'}$.

$$B_i = (00100101)_2 = 25_{hex}$$

Luego,

$$S((C2)_{hex}) = (25)_{hex}$$

Y estas son las salidas que se muestran en la Figura 7.

		y														
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
	0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
	1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
	2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
	3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
	4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
	5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
	6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
	7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
x	8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
	9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
	A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
	B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
	C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
	D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
	E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
	F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Figure: AES S-Box, Valores de sustitución en notación hexadecimal para byte de entradas (xy)

- En el AES, la capa de difusión consiste de dos subcapas:
  - La transformación *ShiftRows*.
  - La transformación *MixColumn*.
- Recordemos que la difusión es la propagación de la influencia de los bits individuales sobre todo el estado.
- A diferencia de los S-box "no lineales". La capa de difusión es una operación lineal de matrices, es decir:

$$DIFF(A + B) = DIFF(A) + DIFF(B).$$

ShiftRows Sublayer:

- La transformación *ShiftRows* es un shifts cíclico, por ejemplo:
- Supongamos que la entrada a la subcapa del ShiftRows es dada por la matriz $B = (B_0, B_1, \ldots, B_{15})$

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

- La salida de la transformación *ShiftRows* es:

B_0	B_4	B_8	B_{12}	$\leftarrow$ no shift
B_5	B_9	B_{13}	B_1	$\leftarrow$ una posición de shift izquierda
B_{10}	B_{14}	B_2	B_6	$\leftarrow$ dos posiciones de shift izquierda
B_{15}	B_3	B_7	B_{11}	$\leftarrow$ tres posiciones de shift izquierda

Advanced Encryption Standard, AES

Capa de Difusión

ShiftRows Sublayer:

La salida de la transformación *ShiftRows* es:

B_0	B_4	B_8	B_{12}	$\leftarrow$ no shift
B_5	B_9	B_{13}	B_1	$\leftarrow$ una posición de shift izquierda
B_{10}	B_{14}	B_2	B_6	$\leftarrow$ dos posiciones de shift izquierda
B_{15}	B_3	B_7	B_{11}	$\leftarrow$ tres posiciones de shift izquierda

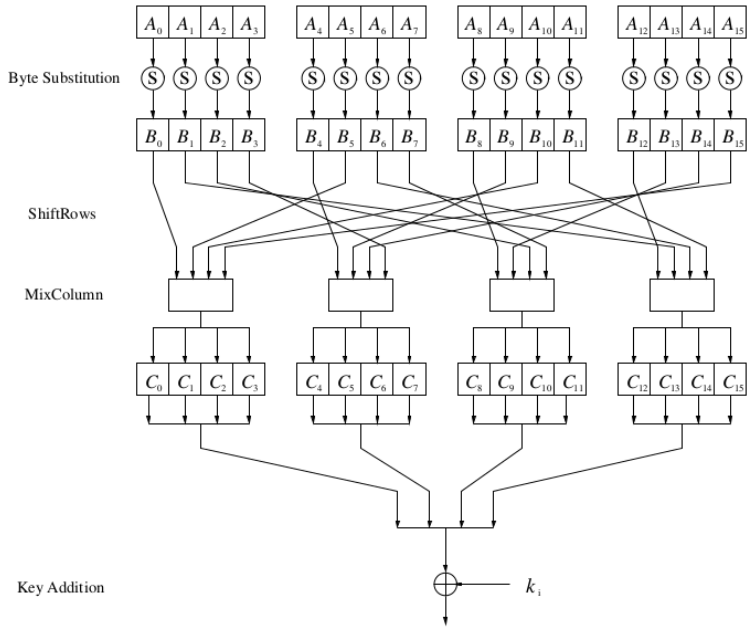


Figure: AES, rondas $1, 2, \ldots, n_r - 1$

MixColumn Sublayer

- La transformación *MixColumn* es lineal y mezcla cada columna de la matriz de estado.
- La idea es que cada byte de entrada tiene influencia en los cuatros bytes de salida.
- La transformación *MixColumn* es la operación que entrega mayor difusión en el AES.
- La combinación de *ShiftRows* y *MixColumn* hace posible que después de tres rondas cada byte de la matriz de estado depende de todos los 16 bytes del plaintex.

MixColumn Sublayer

- La transformación *MixColumn* es lineal y mezcla cada columna de la matriz de estado.
- La idea es que cada byte de entrada tiene influencia en los cuatros bytes de salida.
- La transformación *MixColumn* es la operación que entrega mayor difusión en el AES.
- La combinación de *ShiftRows* y *MixColumn* hace posible que después de tres rondas cada byte de la matriz de estado depende de todos los 16 bytes del plaintex.

MixColumn Sublayer

- La transformación *MixColumn* es lineal y mezcla cada columna de la matriz de estado.
- La idea es que cada byte de entrada tiene influencia en los cuatros bytes de salida.
- La transformación *MixColumn* es la operación que entrega mayor difusión en el AES.
- La combinación de *ShiftRows* y *MixColumn* hace posible que después de tres rondas cada byte de la matriz de estado depende de todos los 16 bytes del plaintex.

MixColumn Sublayer

- La transformación *MixColumn* es lineal y mezcla cada columna de la matriz de estado.
- La idea es que cada byte de entrada tiene influencia en los cuatros bytes de salida.
- La transformación *MixColumn* es la operación que entrega mayor difusión en el AES.
- La combinación de *ShiftRows* y *MixColumn* hace posible que después de tres rondas cada byte de la matriz de estado depende de todos los 16 bytes del plaintex.

MixColumn Sublayer

- Denotemos el 16-byte del estado de entrada B y los 16-byte de estado de salida C:

$$MixColumn(B) = C,$$

donde B es el estado después de la transformación *ShiftRows*.

- Ahora, cada columna de 4-byte se considera como una vector y multiplicado por una matriz fija de 4×4.
- La matriz contiene entradas contantes. La multiplicación y adiciones es realizada en $GF(2^8)$, por ejemplo:

$$\begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} B_0 \\ B_5 \\ B_{10} \\ B_{15} \end{pmatrix}.$$

MixColumn Sublayer

- La segunda columna de los bytes de salida (C_4, C_5, C_6, C_7) es calculada al multiplicar los cuatro bytes de entrada (B_4, B_9, B_{14}, B_3) por la misma matriz de entrada.
- En la figura muestran los bytes de entrada que son usados en cada una de las cuatro columnas de la operación *MixColumn*.

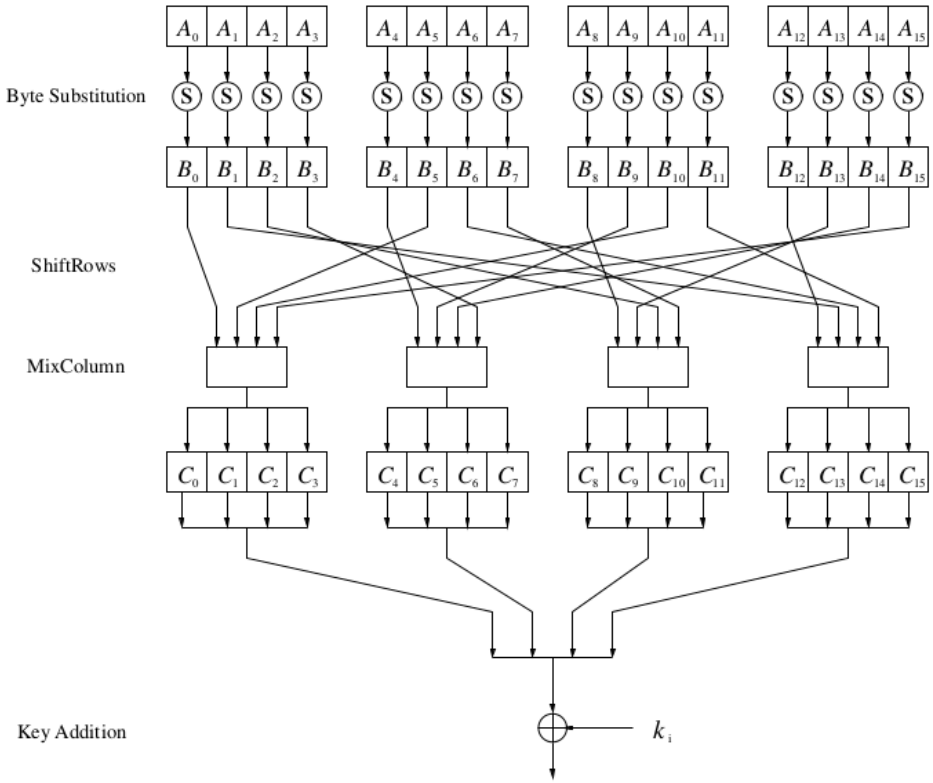


Figure: AES, rondas $1, 2, \ldots, n_r - 1$

MixColumn Sublayer

- Más detalladamente la multiplicación entre el vector y la matriz en esta operación se realiza en $GF(2^8)$ donde cada byte C_i y B_i es un valor de 8-bit en $GF(2^8)$. Los coeficientes de la matriz en notación hexadecimal se utiliza, es decir:
  - 01 se refiere al polinomio con coeficientes (00000001) en $GF(2^8)$ es decir es el elemento 1 en este cuerpo.
  - 02 es el polinomio (00000010) es decir el polinomio x en $GF(2^8)$.
  - 03 se refiere al polinomio con bit (00000011) es decir es el polinomio $x+1$ de $GF(2^8)$.

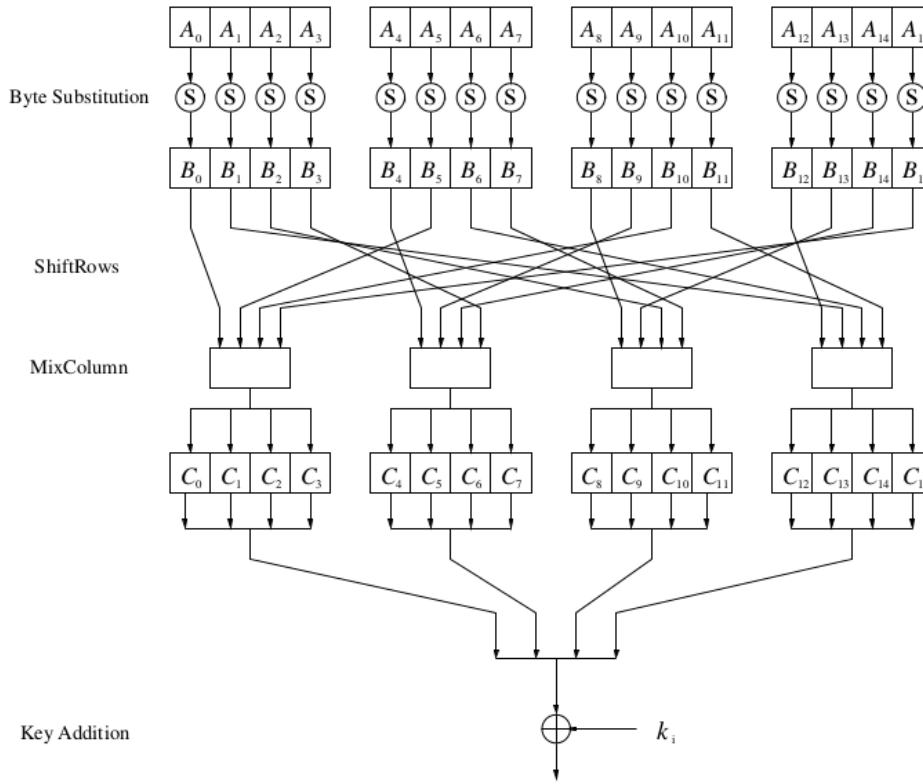


Figure: AES, para las rondas $1, 2, \ldots, n_r - 1$

MixColumn Sublayer

Ejemplo

Asumamos que el vector de entrada a la transformación MixColumn es

$$B = (25, 25, \ldots, 25)$$

En este caso solo dos multiplicaciones son realizadas en $GF(2^8)$. *Estas son* $02 \cdot 25$ *y* $03 \cdot 25$ *en notación polinomial son:*

$$\begin{aligned} 02 \cdot 25 &= x \cdot (x^5 + x^2 + 1) \\ &= x^6 + x^3 + x \end{aligned}$$

$$\begin{aligned} 03 \cdot 25 &= (x + 1) \cdot (x^5 + x^2 + 1) \\ &= (x^6 + x^3 + x) + (x^5 + x^2 + 1) \\ &= x^6 + x^5 + x^3 + x^2 + x + 1 \end{aligned}$$

Ejemplo

No se necesita reducir, es decir, aplicar el polinomio $P(x)$ ya que ambos polinomios tienen grado menor que 8.

Los bytes de salida C del resultado se realiza la siguiente adición en $GF(2^8)$:

$$01 \cdot 25 = x^5 + x^2 + 1$$
$$01 \cdot 25 = x^5 + x^2 + 1$$
$$02 \cdot 25 = x^6 + x^3 + x$$
$$03 \cdot 25 = x^6 + x^5 + x^3 + x^2 + x + 1$$

El resultado es

$$C_i = x^5 + x^2 + 1$$

Donde $i = 0, \ldots, 15$. Esto lleva al estado de salida $C = (25, 25, \ldots, 25)$.

Advanced Encryption Standard, AES

Key Addition Layer

Key Addition Layer

- Las dos entradas de la *Key Addition Layer* son de una matriz de estados de 16-byte y una *subkey* que también consiste de 16 bytes (128-bits).
- Las dos entradas son combinadas a través de una operación XOR bit a bit.
- Notemos que la operación XOR es igual a una adición en el cuerpo de Galois $GF(2)$.
- Las subkey son derivadas en la key-shedule que es descrito en la próxima sección.

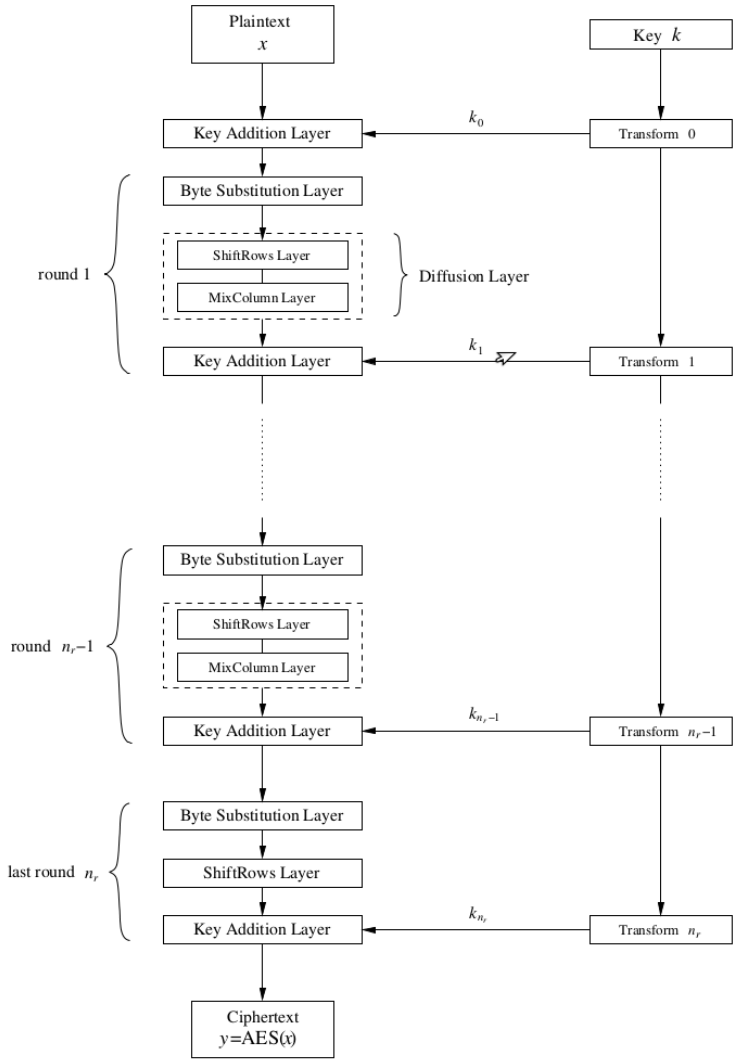


Figure: Diagrama de encriptación del AES

Advanced Encryption Standard, AES

Key Addition Layer

Key Addition Layer

- Las dos entradas de la *Key Addition Layer* son de una matriz de estados de 16-byte y una *subkey* que también consiste de 16 bytes (128-bits).
- Las dos entradas son combinadas a través de una operación XOR bit a bit.
- Notemos que la operación XOR es igual a una adición en el cuerpo de Galois $GF(2)$.
- Las subkey son derivadas en la key-shedule que es descrito en la próxima sección.

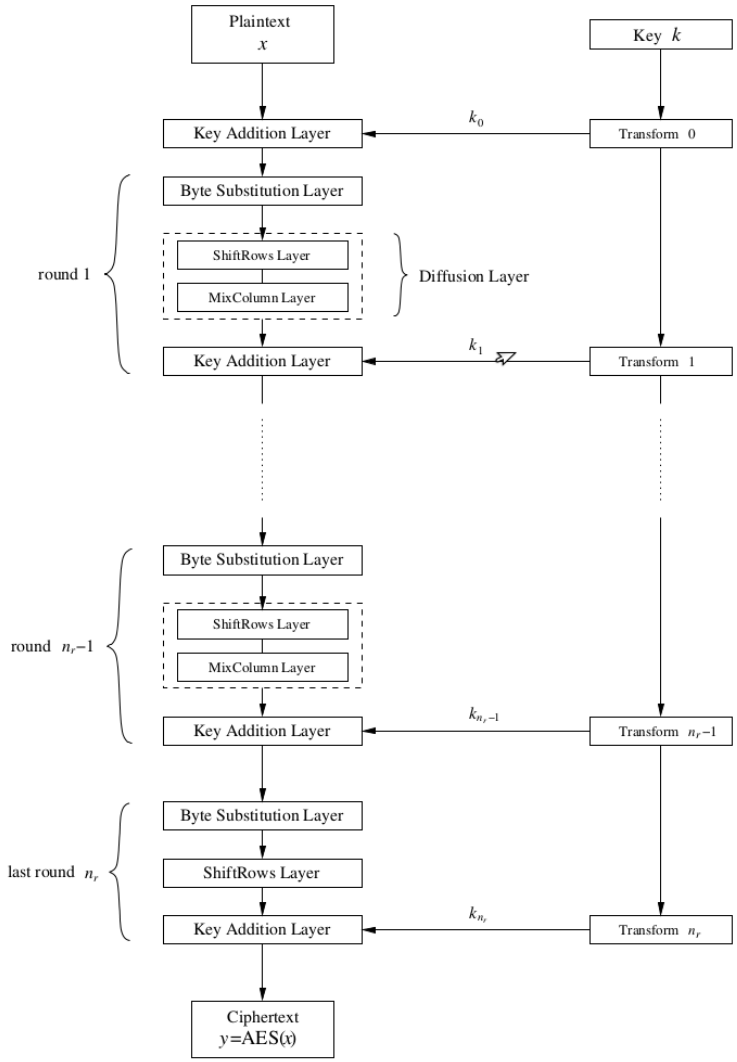


Figure: Diagrama de encriptación del AES

Advanced Encryption Standard, AES

Key Addition Layer

Key Addition Layer

- Las dos entradas de la *Key Addition Layer* son de una matriz de estados de 16-byte y una *subkey* que también consiste de 16 bytes (128-bits).
- Las dos entradas son combinadas a través de una operación XOR bit a bit.
- Notemos que la operación XOR es igual a una adición en el cuerpo de Galois $GF(2)$.
- Las subkey son derivadas en la key-shedule que es descrito en la próxima sección.

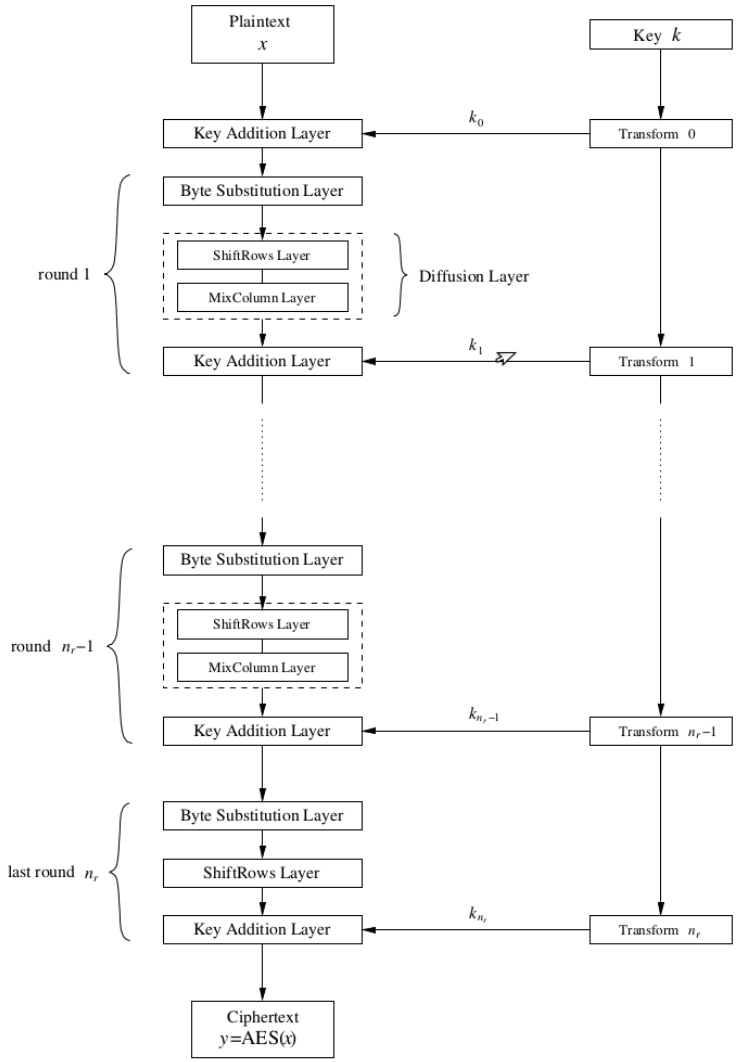


Figure: Diagrama de encriptación del AES

Key Addition Layer

- Las dos entradas de la *Key Addition Layer* son de una matriz de estados de 16-byte y una *subkey* que también consiste de 16 bytes (128-bits).
- Las dos entradas son combinadas a través de una operación XOR bit a bit.
- Notemos que la operación XOR es igual a una adición en el cuerpo de Galois $GF(2)$.
- Las subkey son derivadas en la key-shedule que es descrito en la próxima sección.

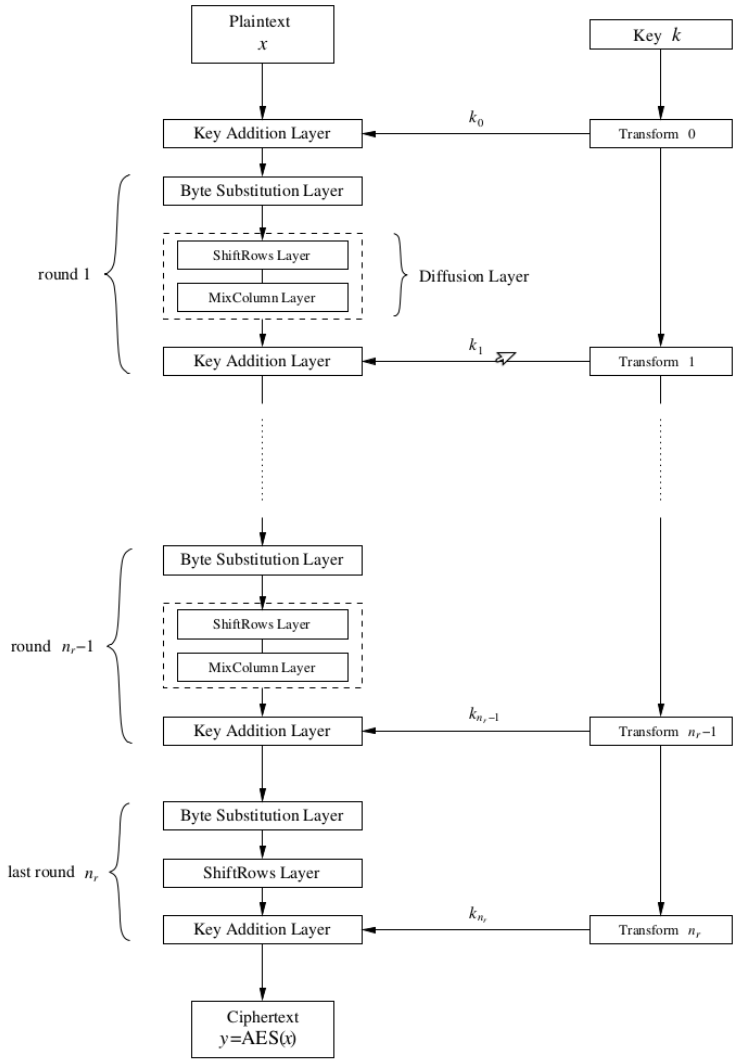


Figure: Diagrama de encriptación del AES

Advanced Encryption Standard, AES

Key Schedule

- La *Key schedule* toma la clave original como entrada de las diferentes tipos de niveles de seguridad (es decir, de 128, 192 o 256 bit de longitud) y deriva las subclaves utilizadas en el AES.
- Observar que una adición XOR de la subclave es usada en la entrada y la salida el AES.
- A veces este proceso se conoce como "whitening".
- El número de subclaves es igual al número de rondas más uno, debido a que las clave necesita para el key whitening en primera *Key Addition Layer*, entonces:

  1) clave de longitud de 128 bits , AES utiliza $10 = n_r$ rondas, entonces necesita 11 subclaves de 128 bits.

  2) clave de longitud de 192 bits , AES utiliza $12 = n_r$ rondas, necesita 13 subclaves de 128 bits .

  3) clave de longitud de 256 bits , AES utiliza $14 = n_r$ rondas, 15 subclaves.

Advanced Encryption Standard, AES

Key Schedule

- La *Key schedule* toma la clave original como entrada de las diferentes tipos de niveles de seguridad (es decir, de 128, 192 o 256 bit de longitud) y deriva las subclaves utilizadas en el AES.
- Observar que una adición XOR de la subclave es usada en la entrada y la salida el AES.
- A veces este proceso se conoce como "whitening".
- El número de subclaves es igual al número de rondas más uno, debido a que las clave necesita para el key whitening en primera *Key Addition Layer*, entonces:

  1) clave de longitud de 128 bits , AES utiliza $10 = n_r$ rondas, entonces necesita 11 subclaves de 128 bits.

  2) clave de longitud de 192 bits , AES utiliza $12 = n_r$ rondas, necesita 13 subclaves de 128 bits .

  3) clave de longitud de 256 bits , AES utiliza $14 = n_r$ rondas, 15 subclaves.

Advanced Encryption Standard, AES

Key Schedule

- La *Key schedule* toma la clave original como entrada de las diferentes tipos de niveles de seguridad (es decir, de 128, 192 o 256 bit de longitud) y deriva las subclaves utilizadas en el AES.
- Observar que una adición XOR de la subclave es usada en la entrada y la salida el AES.
- A veces este proceso se conoce como "whitening".
- El número de subclaves es igual al número de rondas más uno, debido a que las clave necesita para el key whitening en primera *Key Addition Layer*, entonces:

  1) clave de longitud de 128 bits , AES utiliza $10 = n_r$ rondas, entonces necesita 11 subclaves de 128 bits.

  2) clave de longitud de 192 bits , AES utiliza $12 = n_r$ rondas, necesita 13 subclaves de 128 bits .

  3) clave de longitud de 256 bits , AES utiliza $14 = n_r$ rondas, 15 subclaves.

Advanced Encryption Standard, AES

Key Schedule

- La *Key schedule* toma la clave original como entrada de las diferentes tipos de niveles de seguridad (es decir, de 128, 192 o 256 bit de longitud) y deriva las subclaves utilizadas en el AES.
- Observar que una adición XOR de la subclave es usada en la entrada y la salida el AES.
- A veces este proceso se conoce como "whitening".
- El número de subclaves es igual al número de rondas más uno, debido a que las clave necesita para el key whitening en primera *Key Addition Layer*, entonces:

  1) clave de longitud de 128 bits , AES utiliza $10 = n_r$ rondas, entonces necesita 11 subclaves de 128 bits.

  2) clave de longitud de 192 bits , AES utiliza $12 = n_r$ rondas, necesita 13 subclaves de 128 bits .

  3) clave de longitud de 256 bits , AES utiliza $14 = n_r$ rondas, 15 subclaves.

Advanced Encryption Standard, AES

Key Schedule

- La *Key schedule* toma la clave original como entrada de las diferentes tipos de niveles de seguridad (es decir, de 128, 192 o 256 bit de longitud) y deriva las subclaves utilizadas en el AES.
- Observar que una adición XOR de la subclave es usada en la entrada y la salida el AES.
- A veces este proceso se conoce como "whitening".
- El número de subclaves es igual al número de rondas más uno, debido a que las clave necesita para el key whitening en primera *Key Addition Layer*, entonces:

  1) clave de longitud de 128 bits , AES utiliza $10 = n_r$ rondas, entonces necesita 11 subclaves de 128 bits.

  2) clave de longitud de 192 bits , AES utiliza $12 = n_r$ rondas, necesita 13 subclaves de 128 bits .

  3) clave de longitud de 256 bits , AES utiliza $14 = n_r$ rondas, 15 subclaves.

Advanced Encryption Standard, AES

Key Schedule

- La *Key schedule* toma la clave original como entrada de las diferentes tipos de niveles de seguridad (es decir, de 128, 192 o 256 bit de longitud) y deriva las subclaves utilizadas en el AES.
- Observar que una adición XOR de la subclave es usada en la entrada y la salida el AES.
- A veces este proceso se conoce como "whitening".
- El número de subclaves es igual al número de rondas más uno, debido a que las clave necesita para el key whitening en primera *Key Addition Layer*, entonces:

  1) clave de longitud de 128 bits , AES utiliza $10 = n_r$ rondas, entonces necesita 11 subclaves de 128 bits.

  2) clave de longitud de 192 bits , AES utiliza $12 = n_r$ rondas, necesita 13 subclaves de 128 bits .

  3) clave de longitud de 256 bits , AES utiliza $14 = n_r$ rondas, 15 subclaves.

Advanced Encryption Standard, AES

Key Schedule

- La *Key schedule* toma la clave original como entrada de las diferentes tipos de niveles de seguridad (es decir, de 128, 192 o 256 bit de longitud) y deriva las subclaves utilizadas en el AES.
- Observar que una adición XOR de la subclave es usada en la entrada y la salida el AES.
- A veces este proceso se conoce como "whitening".
- El número de subclaves es igual al número de rondas más uno, debido a que las clave necesita para el key whitening en primera *Key Addition Layer*, entonces:

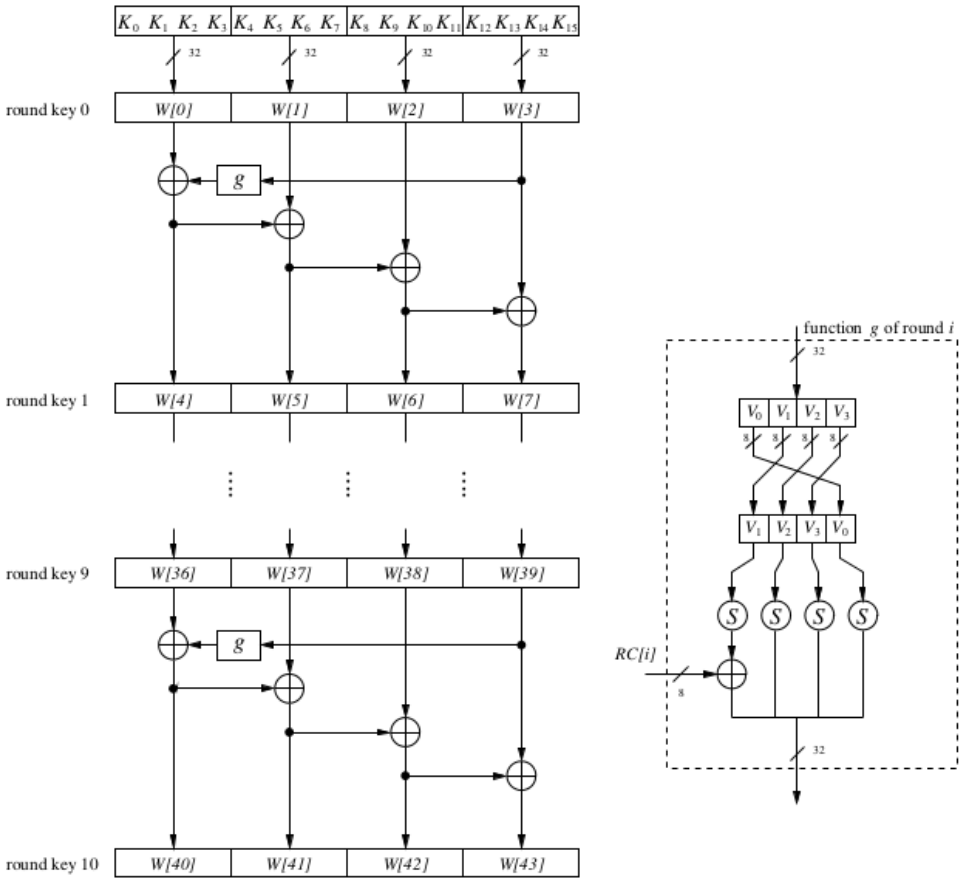
  1) clave de longitud de 128 bits , AES utiliza $10 = n_r$ rondas, entonces necesita 11 subclaves de 128 bits.
  2) clave de longitud de 192 bits , AES utiliza $12 = n_r$ rondas, necesita 13 subclaves de 128 bits .
  3) clave de longitud de 256 bits , AES utiliza $14 = n_r$ rondas, 15 subclaves.

Key Schedule

- La subkey son calculada de manera recursiva, es decir, para calcular k_i se necesita conocer la subclave k_{i-1}.
- En el AES el key shedule trabaja orientada a palabras, donde 1 palabra es de igual a 32bits.
- Las subclaves son almacenadas en un arreglo W que consiste en palabras.
- Existen diferentes key schedules para los tres diferentes tamaños de las claves (128, 192, 256 bit).

Key Schedule para 128-bit key AES

- Las 11 subclaves son almacenadas en un arreglo de clave expandido con los elementos $W[0], \ldots, W[43]$.
- Las subclaves son calculadas como se ve en la figura.
- Los elementos $K_0, \ldots, K_{15}$ denota los bytes originales de la clave del AES.



Key Schedule para 128-bit key AES

- Observemos que la primera subkey k_0 es el original AES key, es decir, la clave es copiada en los primeros cuatro elementos de la arreglo de la clave W.
- Los otros elementos del arreglo son calculados como se puede ver en la figura.
- La palabra más a la izquierda de una subclave $W[4i]$ donde $i = 1, \ldots, 10$. es calculado como:

$$W[4i] = W[4(i-1)] + g(W[4i-1]).$$

Donde $g()$ es una función lineal con cuatro-byte de entrada y salida.

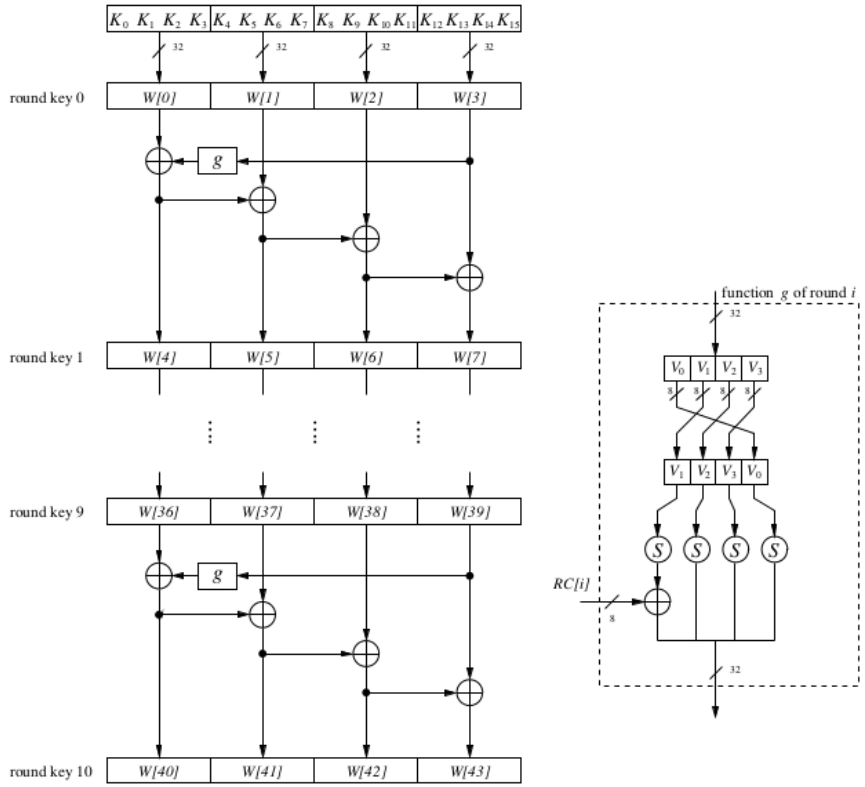


Figure: AES key schedule para clave de 128 bits.

Key Schedule para 128-bit key AES

- Las demás tres palabras de la subclave son calculadas de manera recursiva como:

$$W[4i+j] = W[4i+j-1]+W[4(i-1)+j],$$

donde $i = 1, \ldots, 10$ y $j = 1, 2, 3$.

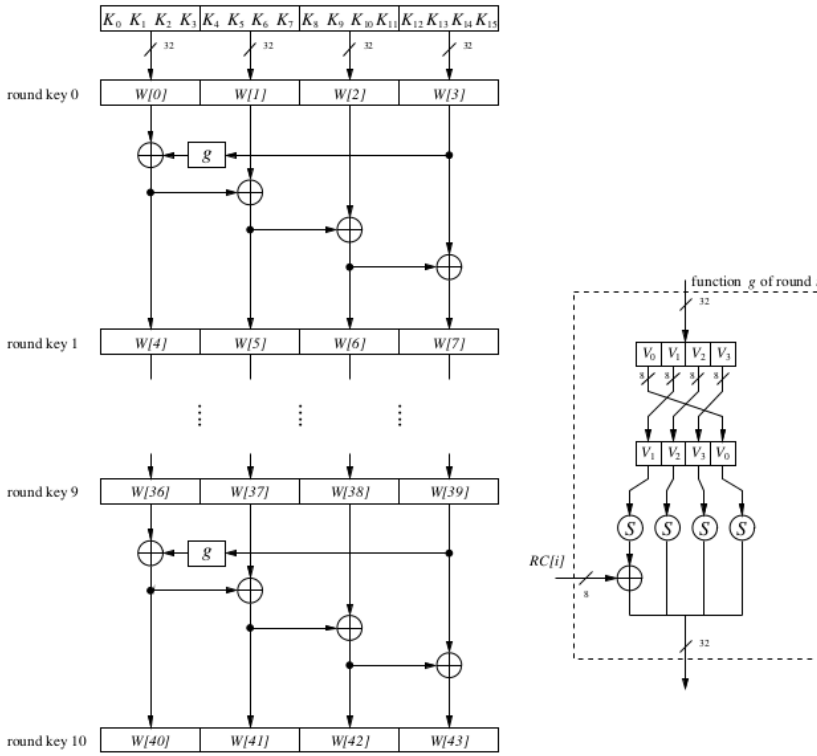


Figure: AES key schedule para clave de tamaño de 128 bits.

Key Schedule para 128-bit key AES

- La función $g()$ gira los cuatro bytes de entrada y calcula byte-a-byte con una sustitución S-Box y añade un *round coefficient RC* a esto.
- El *round coefficient* es un elemento del cuerpo de Galois $GF(2^8)$, es decir, un valor de 8-bit.
- Este es solo añadido el los bytes más a la izquierda en la función $g()$.

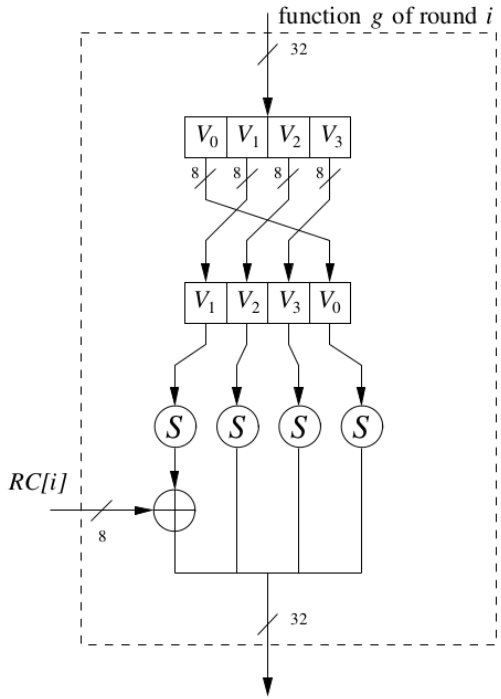


Figure: Función $g()$.

Key Schedule para 128-bit key AES

- El *round coefficient* varía desde ronda a ronda de acuerdo a la siguiente regla:

$$RC[1] = x^0 = (00000001)_2$$
$$RC[2] = x^1 = (00000010)_2$$
$$RC[3] = x^2 = (00000100)_2$$
$$\vdots = \vdots = \vdots$$
$$RC[10] = x^9 = (00110110)_2$$

- La función $g()$ tiene dos propósitos:
  1. Primero, es añadir no-linealidad al *key schedule*.
  2. Segundo, remueve la simetría del AES. Ambas propiedades son necesarias para frustrar ciertos ataques de block cipher.

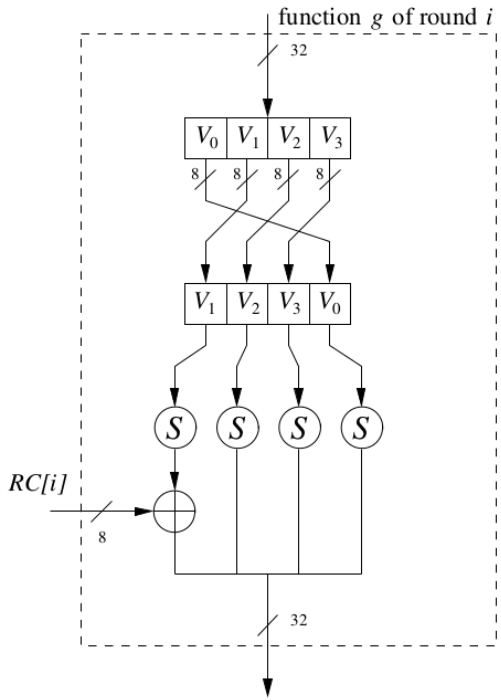


Figure: Función $g()$.

Advanced Encryption Standard, AES

Key Schedule para 192-Bit Key AES

- AES con clave de 192-bit utiliza 12 rondas y por lo tanto 13 subclaves de 128 bit cada una.
- Las subclaves requieren 52 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[51]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la figura.

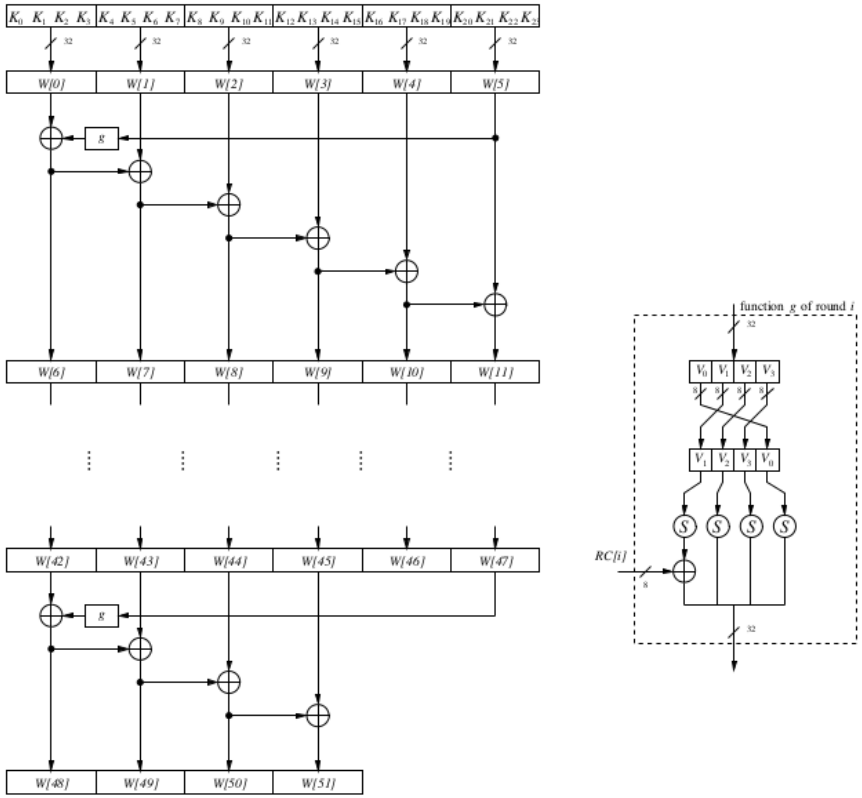


Figure: AES key schedule para clave de tamaño de 192 bits.

Advanced Encryption Standard, AES

Key Schedule para 192-Bit Key AES

- Hay 8 iteraciones del key schedule. Cada iteración calcula seis nuevas palabras de la subclave del arreglo W.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$.
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las ocho rondas de los coeficientes $RC[i]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit y rangos desde $RC[1], \ldots, RC[8]$.

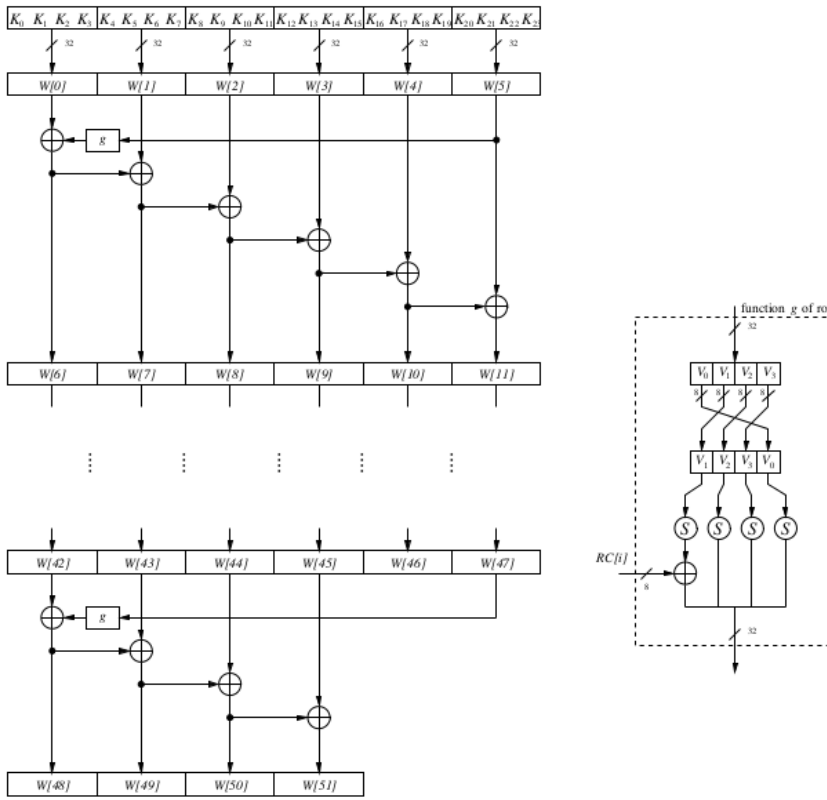


Figure: AES key schedule para clave de tamañ de 192 bits.

Advanced Encryption Standard, AES

Key Schedule para 192-Bit Key AES

- Hay 8 iteraciones del key schedule. Cada iteración calcula seis nuevas palabras de la subclave del arreglo W.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$.
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las ocho rondas de los coeficientes $RC[i]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit y rangos desde $RC[1], \ldots, RC[8]$.

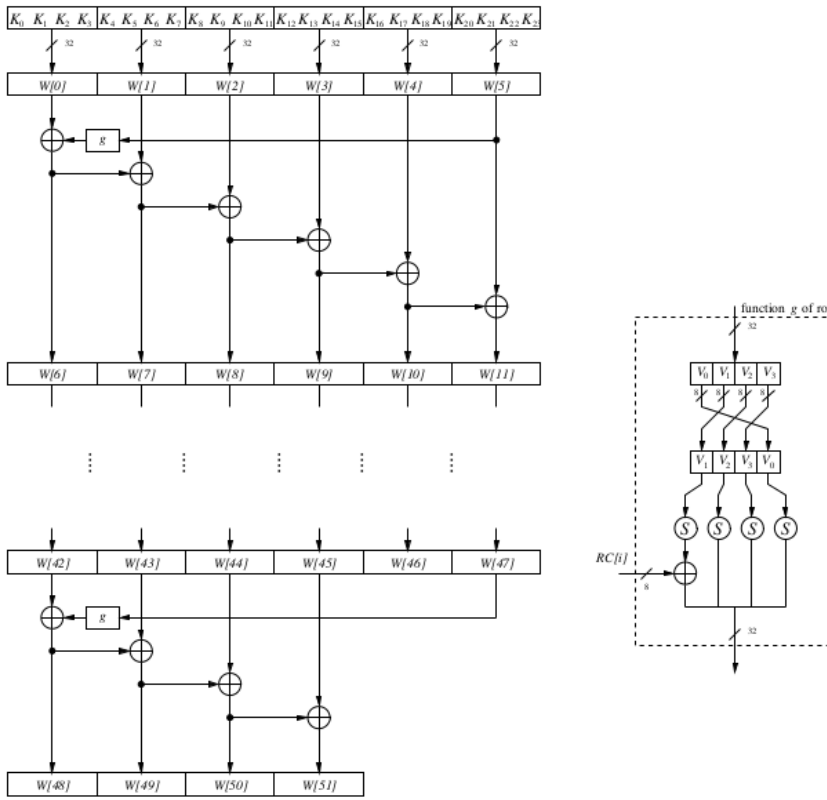


Figure: AES key schedule para clave de tamañ de 192 bits.

Advanced Encryption Standard, AES

Key Schedule para 192-Bit Key AES

- Hay 8 iteraciones del key schedule. Cada iteración calcula seis nuevas palabras de la subclave del arreglo W.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$.
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las ocho rondas de los coeficientes $RC[i]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit y rangos desde $RC[1], \ldots, RC[8]$.

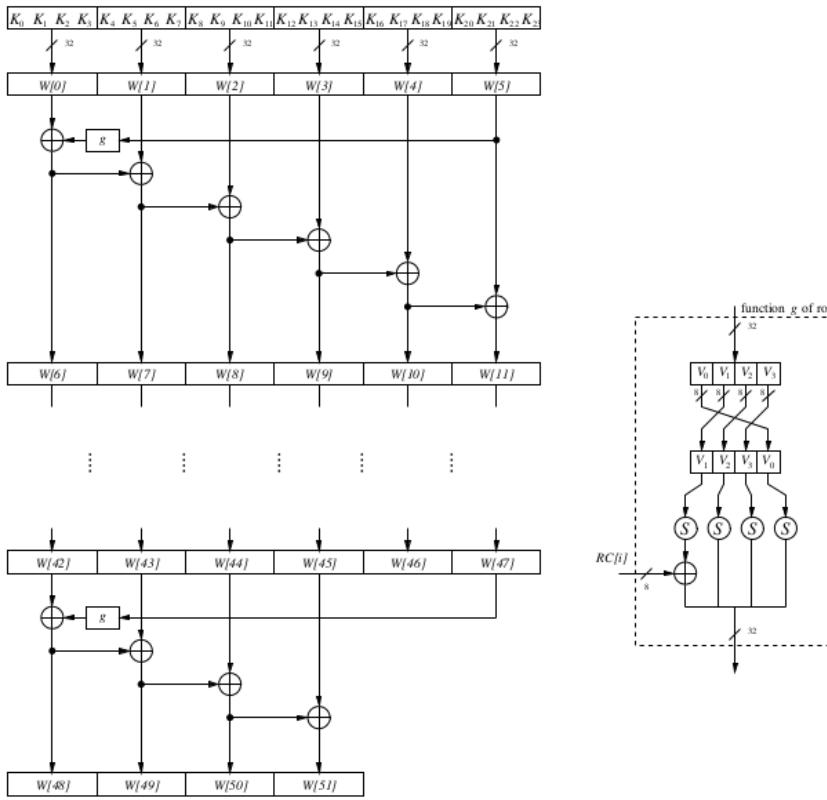


Figure: AES key schedule para clave de tamañ de 192 bits.

Advanced Encryption Standard, AES

Key Schedule para 192-Bit Key AES

- Hay 8 iteraciones del key schedule. Cada iteración calcula seis nuevas palabras de la subclave del arreglo W.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$.
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las ocho rondas de los coeficientes $RC[i]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit y rangos desde $RC[1], \ldots, RC[8]$.

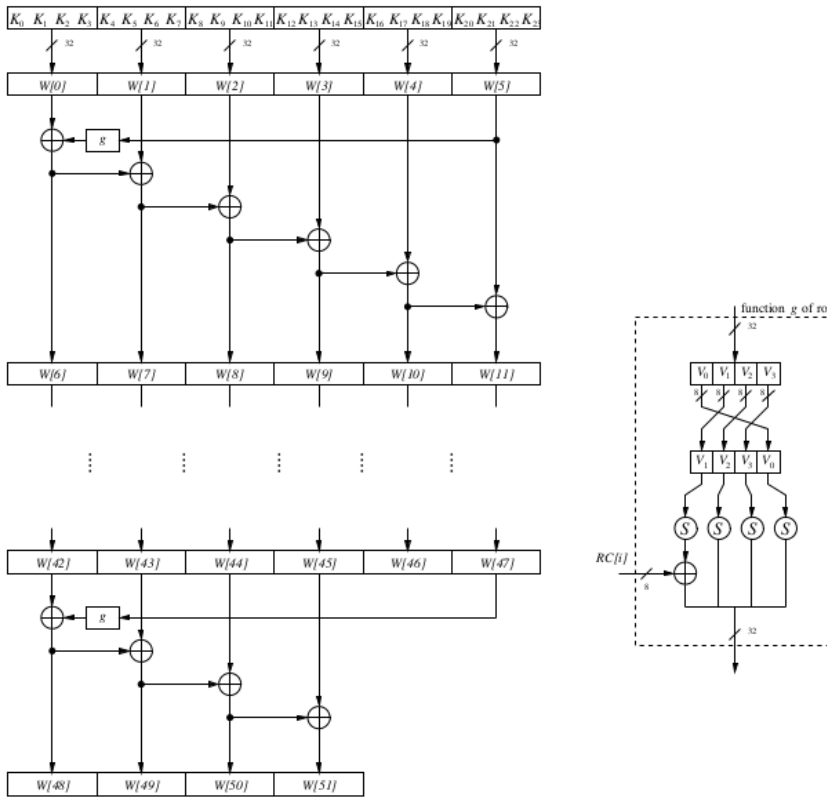


Figure: AES key schedule para clave de tamañ de 192 bits.

Advanced Encryption Standard, AES

Key Schedule para 192-Bit Key AES

- Hay 8 iteraciones del key schedule. Cada iteración calcula seis nuevas palabras de la subclave del arreglo W.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$.
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las ocho rondas de los coeficientes $RC[i]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit y rangos desde $RC[1], \ldots, RC[8]$.

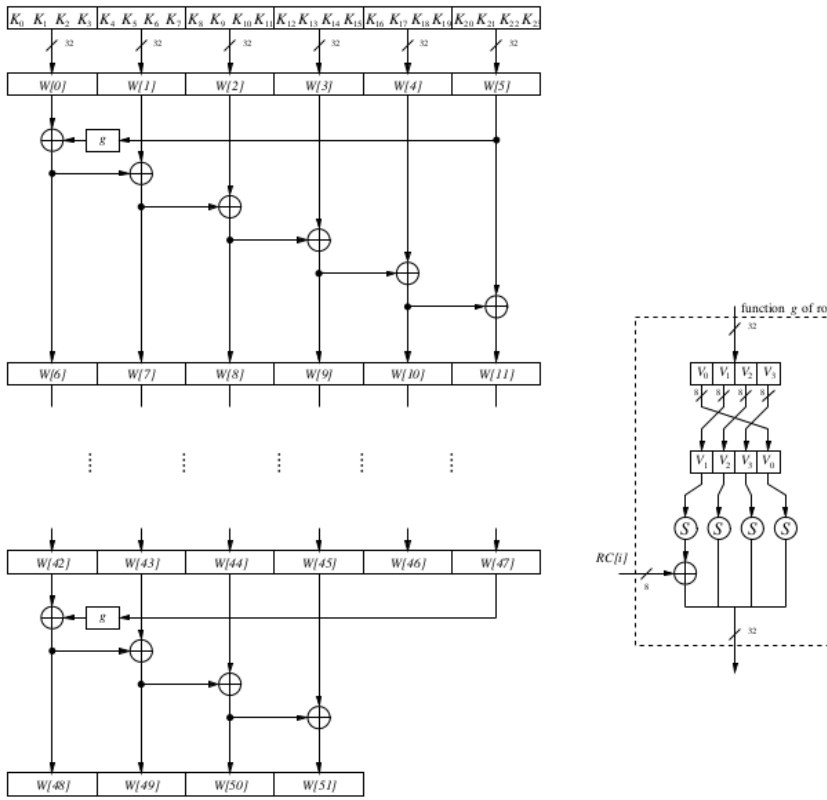


Figure: AES key schedule para clave de tamañ de 192 bits.

Advanced Encryption Standard, AES

Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.

Advanced Encryption Standard, AES

Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.

Advanced Encryption Standard, AES

Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.

Advanced Encryption Standard, AES

Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.

Advanced Encryption Standard, AES

Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.

Advanced Encryption Standard, AES

Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.

Advanced Encryption Standard, AES

Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.

Advanced Encryption Standard, AES

Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.

Advanced Encryption Standard, AES

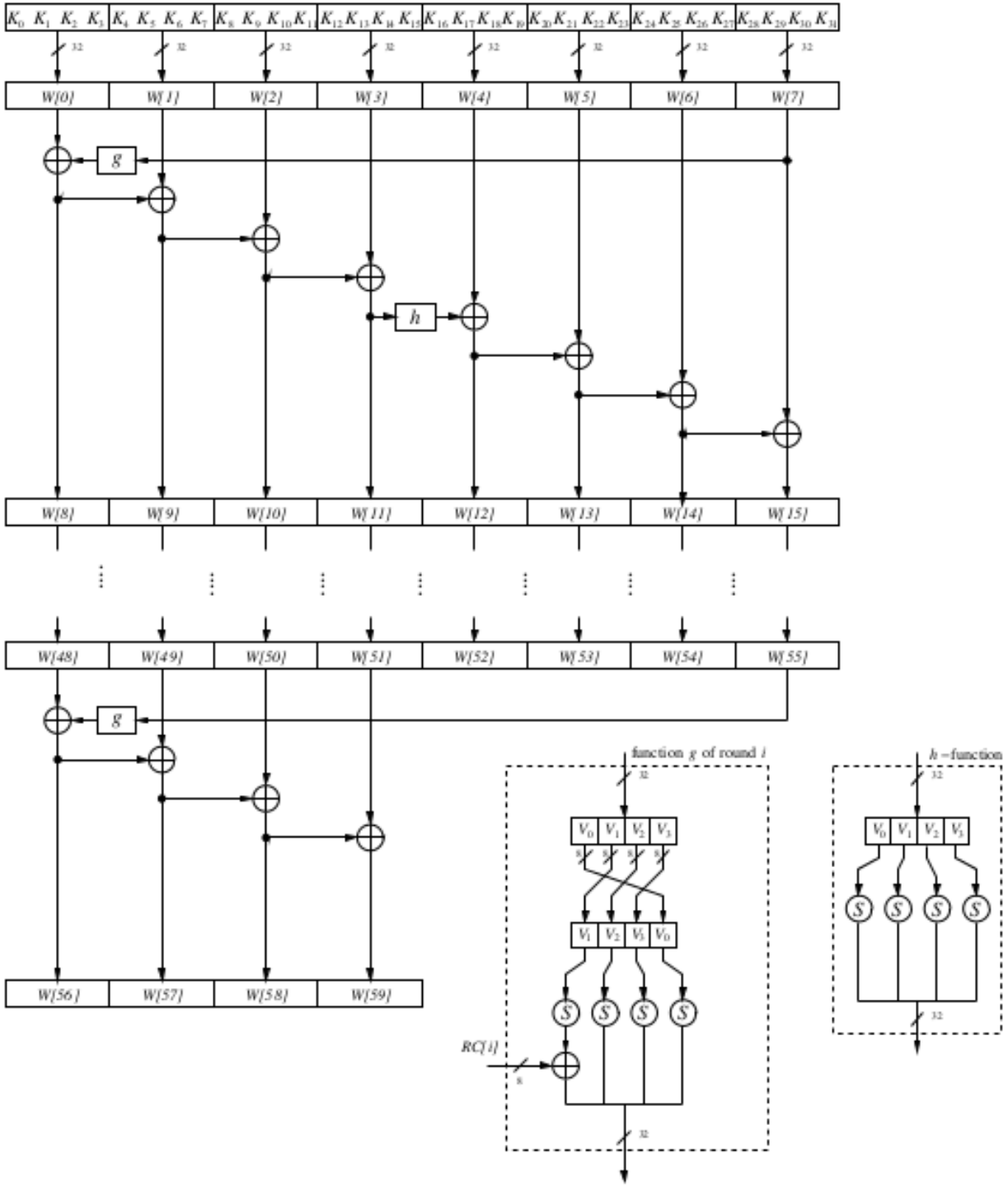
Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.

Advanced Encryption Standard, AES

Key Schedule for 256-Bit Key AES

- AES con clave de 256-bit utiliza 15 subclaves de 128 bit cada una.
- Las subclaves son almacenadas en 60 palabras, que son almacenadas en un arreglo de elementos $W[0], \ldots, W[59]$.
- EL cálculo de los elementos del arreglo es similar al caso del AES con claves de 128-bit y como se puede ver en la siguiente figura.
- El key schedule tiene siete iteraciones, donde cada iteración calcula ocho palabras para la subclave.
- La subclave para la primera ronda del AES es formada por los elementos del arreglo $(W[0], W[1], W[2], W[3])$
- El segundo subclave por los elementos $(W[4], W[5], W[6], W[7])$ y así en adelante.
- Las siete rondas de los coeficientes $RC[1], \ldots, RC[7]$ se necesitan dentro de la función $g()$.
- Se calculan con el caso de 128-bit.
- Este key schedule también tiene una función $h()$ con 4-byte de entrada y de salida.
- La función aplica el S-Box para todos los cuatro bytes de entrada.


W[0]
W[1]
W[2]
W[3]
W[4]
W[5]
W[6]
W[7]
g
h
W[8]
W[9]
W[10]
W[11]
W[12]
W[13]
W[14]
W[15]
W[48]
W[49]
W[50]
W[51]
W[52]
W[53]
W[54]
W[55]
W[56]
W[57]
W[58]
W[59]
function g of round i
RC[i]
h –function

Advanced Encryption Standard, AES

En general, cuando se implementa cualquier key schedules, dos diferentes enfoques existen:

Pre-cálculos

- Todas las subclaves son expandidas primero en el arreglo W.
- La encriptación y (desencriptación) del plaintex (cibertext) son ejecutadas después.
- Este enfoque es utilizado en implementaciones del AES en PC o en servidores, donde grandes cantidades de información son encriptadas con una clave.
- Observemos que este enfoque necesita $(n_r + 1)\dot{1}6$ bytes de memoria, es decir, de $11\dot{1}6 = 176$ bytes si la clave utilizada es de 128 bits.
- Esta es la razón por la cual este enfoque no se utiliza para implementaciones con recursos limitados de memoria como lo son las smart card.

Advanced Encryption Standard, AES

En general, cuando se implementa cualquier key schedules, dos diferentes enfoques existen:

Pre-cálculos

- Todas las subclaves son expandidas primero en el arreglo W.
- La encriptación y (desencriptación) del plaintex (cibertext) son ejecutadas después.
- Este enfoque es utilizado en implementaciones del AES en PC o en servidores, donde grandes cantidades de información son encriptadas con una clave.
- Observemos que este enfoque necesita $(n_r + 1)\dot{1}6$ bytes de memoria, es decir, de $11\dot{1}6 = 176$ bytes si la clave utilizada es de 128 bits.
- Esta es la razón por la cual este enfoque no se utiliza para implementaciones con recursos limitados de memoria como lo son las smart card.

Advanced Encryption Standard, AES

En general, cuando se implementa cualquier key schedules, dos diferentes enfoques existen:

Pre-cálculos

- Todas las subclaves son expandidas primero en el arreglo W.
- La encriptación y (desencriptación) del plaintex (cibertext) son ejecutadas después.
- Este enfoque es utilizado en implementaciones del AES en PC o en servidores, donde grandes cantidades de información son encriptadas con una clave.
- Observemos que este enfoque necesita $(n_r + 1)\dot{1}6$ bytes de memoria, es decir, de $11\dot{1}6 = 176$ bytes si la clave utilizada es de 128 bits.
- Esta es la razón por la cual este enfoque no se utiliza para implementaciones con recursos limitados de memoria como lo son las smart card.

Advanced Encryption Standard, AES

En general, cuando se implementa cualquier key schedules, dos diferentes enfoques existen:

Pre-cálculos

- Todas las subclaves son expandidas primero en el arreglo W.
- La encriptación y (desencriptación) del plaintex (cibertext) son ejecutadas después.
- Este enfoque es utilizado en implementaciones del AES en PC o en servidores, donde grandes cantidades de información son encriptadas con una clave.
- Observemos que este enfoque necesita $(n_r + 1)\dot{1}6$ bytes de memoria, es decir, de $11\dot{1}6 = 176$ bytes si la clave utilizada es de 128 bits.
- Esta es la razón por la cual este enfoque no se utiliza para implementaciones con recursos limitados de memoria como lo son las smart card.

Advanced Encryption Standard, AES

En general, cuando se implementa cualquier key schedules, dos diferentes enfoques existen:

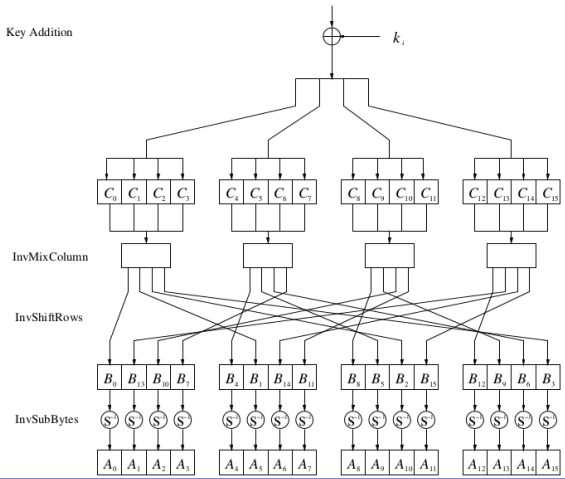
Pre-cálculos

- Todas las subclaves son expandidas primero en el arreglo W.
- La encriptación y (desencriptación) del plaintex (cibertext) son ejecutadas después.
- Este enfoque es utilizado en implementaciones del AES en PC o en servidores, donde grandes cantidades de información son encriptadas con una clave.
- Observemos que este enfoque necesita $(n_r + 1)\dot{1}6$ bytes de memoria, es decir, de $11\dot{1}6 = 176$ bytes si la clave utilizada es de 128 bits.
- Esta es la razón por la cual este enfoque no se utiliza para implementaciones con recursos limitados de memoria como lo son las smart card.

Advanced Encryption Standard, AES

On the fly

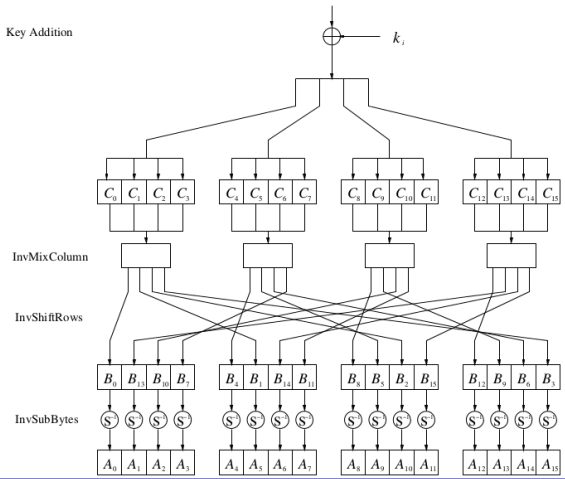
- Una nueva subclave es derivada para cada nueva ronda durante la encriptación (desencriptación) del plaintext (ciphertext).
- Se debe observar que al descifrar el ciphertext, la última subclave es XOR con primer ciphertext.
- Por lo tanto, se requiere para derivar de forma recursiva todas las subclaves primero y luego, comenzar con la desencriptación del ciphertext y la generación en la marcha de subclaves.
- Como resultado de esta sobrecarga, la desencriptación de un texto cifrado es siempre ligeramente más lento que el cifrado de un plaintext cuando se utiliza la generación de subclaves en la marcha.



Advanced Encryption Standard, AES

On the fly

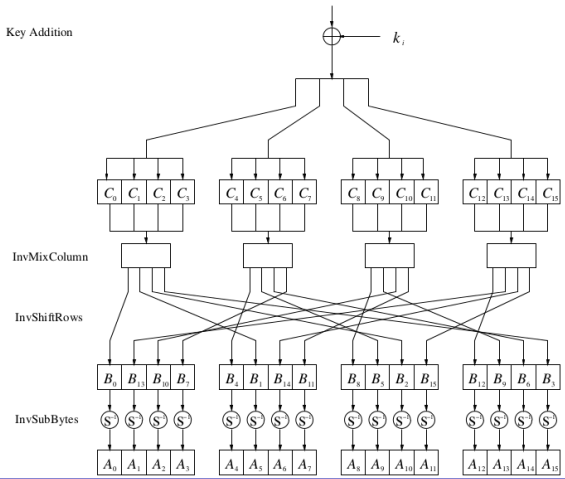
- Una nueva subclave es derivada para cada nueva ronda durante la encriptación (desencriptación) del plaintext (ciphertext).
- Se debe observar que al descifrar el ciphertext, la última subclave es XOR con primer ciphertext.
- Por lo tanto, se requiere para derivar de forma recursiva todas las subclaves primero y luego, comenzar con la desencriptación del ciphertext y la generación en la marcha de subclaves.
- Como resultado de esta sobrecarga, la desencriptación de un texto cifrado es siempre ligeramente más lento que el cifrado de un plaintext cuando se utiliza la generación de subclaves en la marcha.



Advanced Encryption Standard, AES

On the fly

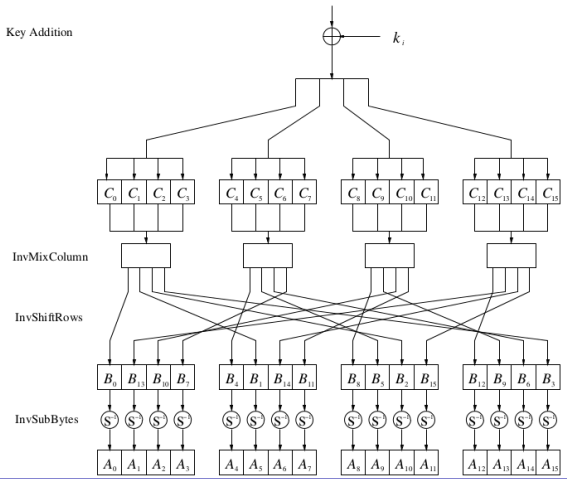
- Una nueva subclave es derivada para cada nueva ronda durante la encriptación (desencriptación) del plaintext (ciphertext).
- Se debe observar que al descifrar el ciphertext, la última subclave es XOR con primer ciphertext.
- Por lo tanto, se requiere para derivar de forma recursiva todas las subclaves primero y luego, comenzar con la desencriptación del ciphertext y la generación en la marcha de subclaves.
- Como resultado de esta sobrecarga, la desencriptación de un texto cifrado es siempre ligeramente más lento que el cifrado de un plaintext cuando se utiliza la generación de subclaves en la marcha.



Advanced Encryption Standard, AES

On the fly

- Una nueva subclave es derivada para cada nueva ronda durante la encriptación (desencriptación) del plaintext (ciphertext).
- Se debe observar que al descifrar el ciphertext, la última subclave es XOR con primer ciphertext.
- Por lo tanto, se requiere para derivar de forma recursiva todas las subclaves primero y luego, comenzar con la desencriptación del ciphertext y la generación en la marcha de subclaves.
- Como resultado de esta sobrecarga, la desencriptación de un texto cifrado es siempre ligeramente más lento que el cifrado de un plaintext cuando se utiliza la generación de subclaves en la marcha.



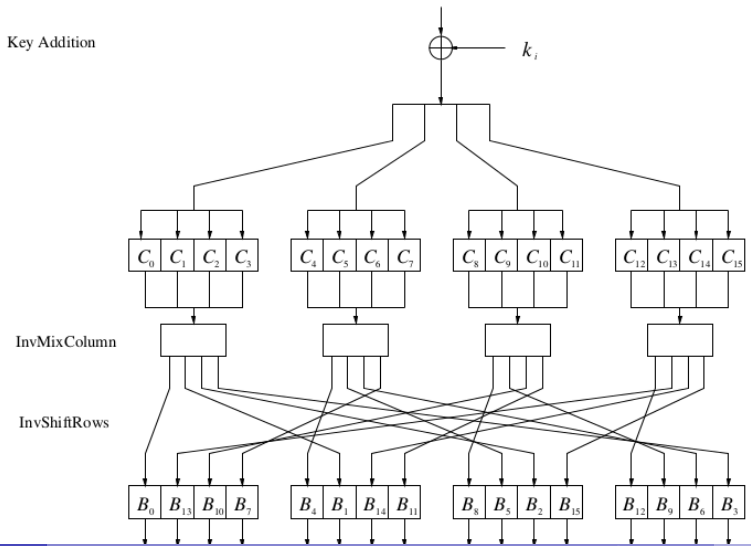
Decryption Advanced Encryption Standard, AES

Dado que el AES no se basa en una red de Feistel, todas las capas en realidad deben ser invertidas, es decir:

1. La capa *Byte Substitution Layer* se convierte en la capa de *Inv Byte-Substitution layer*.
2. La capa *ShiftRows layer* se convierte en la capa *Inv ShiftRows layer*, y la capa *MixColumn layer* se convierte en capa *Inv-MixColumn*.

Sin embargo, las operaciones de capa inversas son bastante similares a las operaciones de capas utilizadas para el cifrado.

Además, el orden de las subclaves se invierte, es decir, necesitamos un key schedule invertido. En la siguiente figura 14 se muestra en diagrama desencriptación del AES.



Decryption Advanced Encryption Standard, AES

Dado que la última ronda de cifrado no realiza la operación *MixColum*, el primera ronda de descifrado también no contiene la capa inversa correspondiente. Sin embardo todas las otras rondas de descifrado contienen todas las capas que utilizan el AES.
A continuación, se discuten las capas inversos del general de AES descifrado redonda (Fig. 15).

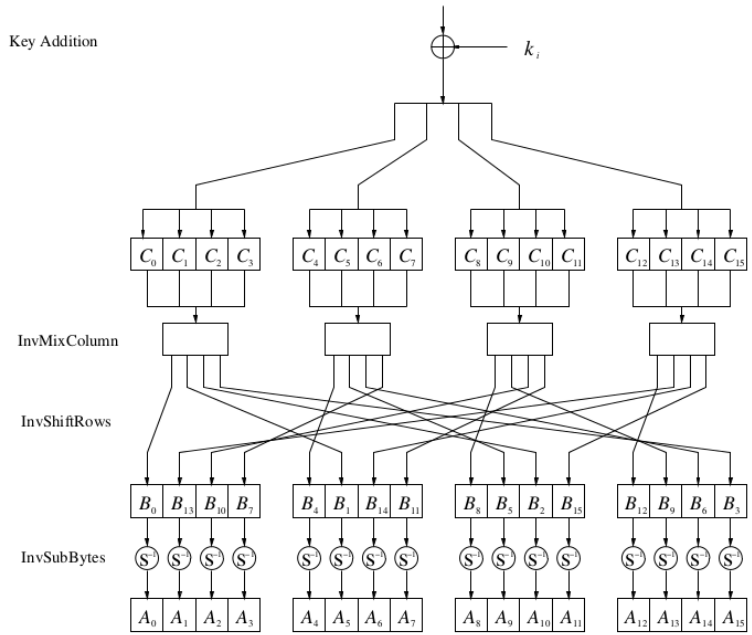


Figure: Desencriptación AES rondas $1, 2, \ldots, n_r - 1$.

Decryption Advanced Encryption Standard, AES

Dado que la operación XOR es su propia inversa, la capa adición clave en el modo de descifrado es el mismo que en el modo de cifrado: consiste en una fila de puertas XOR.

Subcapa MixColumn Inverso

Después de la adición de la subclave, el paso *Inverso MixColumn* se aplica para el estado (una vez más, la excepción es la primera ronda de descifrado). Con el fin de revertir la operación *MixColumn*, se debe utilizar la inversa de su matriz. La entrada es una columna de 4-bytes del Estado C, que se multiplica por la matriz inversa de 4×4. La matriz contiene entradas constantes.

Decryption Advanced Encryption Standard, AES

La multiplicación y la adición de los coeficientes se realiza en $GF(2^8)$.

$$\begin{pmatrix} B_0 \\ B_1 \\ B_2 \\ B_3 \end{pmatrix} = \begin{pmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{pmatrix} \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix}.$$

La segunda columna de los bytes de salida (B_4, B_5, B_6, B_7) es calculado multiplicando la misma matriz constante por los cuatros bytes de entrada (C_4, C_5, C_6, C_7) y así en adelante. Todos los elementos B_i, C_i y elementos constantes de la matriz inversa son elementos en $GF(2^8)$. Por ejemplo

$$0B = (0B)_{hex} = (00001011)_2 = x^3 + x + 1.$$

La suma en la multiplicación entre vector y matriz es un XOR bit a bit.

Inverse ShiftRows Sublayer

Con el fin de revertir la operación *ShiftRows* del algoritmo de cifrado, debemos desplazar las filas de la matriz de estado en la dirección opuesta.

La primera fila no se cambia por la transformación *ShiftRows* inversa. Si la entrada de la subcapa *ShiftRows* se administra como una matriz de estado $B = (B_0, B_1, \ldots, B_{15})$:

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

la subcapa ShiftRows inversa produce la salida:

B_0	B_4	B_8	B_{12}	$\rightarrow$ no shift
B_{13}	B_1	B_5	B_9	$\rightarrow$ una posición de shift derecha
B_{10}	B_{14}	B_2	B_6	$\rightarrow$ dos posiciones de shift derecha
B_7	B_{11}	B_{15}	B_3	$\rightarrow$ tres posiciones de shift derecha

Decryption Advanced Encryption Standard, AES

Inverso Byte Substitution Layer

La S-Box inversa es lo que se utiliza al descifrar un texto cifrado. Dado que la AES S-Box es una biyectiva, es decir, un uno-a-uno, es posible construir un S-Box inversa de tal manera que:

$$A_i = S^{-1}(B_i) = S^{-1}(S(A_i))$$

donde A_i y B_i son elementos de la matriz estado.

La entradas del S-Box inverso son dadas en la siguiente figura:

x \ y	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

Decryption Advanced Encryption Standard, AES

$$\begin{pmatrix} b_0' \\ b_1' \\ b_2' \\ b_3' \\ b_4' \\ b_5' \\ b_6' \\ b_7' \end{pmatrix} \equiv \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \end{pmatrix} \mod 2$$

donde los $(b_7, \ldots, b_0)$ es una representación vectorial bit a bit de $B_i(x)$, y $(b_7', \ldots b_0')$ es el resultado después de la transformación affine.

El segundo paso de la operación del S-box inverso, es un inverso en el cuerpo de Galois que debe ser invertido. Para esto, notar que $A_i = (A_i^{-1})^{-1}$. Esto significa que la operación inverso es revertida por el calculo de la inversa nuevamente. Para la notación se debe calcular

$$A_i = (B_i^{'})^{-1} \in GF(2^8)$$

con el polinomio fijo de reducción $P(x) = x^8 + x^4 + x^3 + x + 1$. Nuevamente el elemento cero es llevado al cero. El vector $A_i = (a_7, \ldots, a_0)$ (representa el elemento del cuerpo $a_7 x^7 + \cdots + a_0$) es el resultado de la sustitución:

$$A_i = S^{-1}(B_i).$$

Key Schedule Decryption

Dado que la primera ronda de desencriptación se necesita la última subclave, en la segunda ronda de descifrado se necesita de la penltima subclave y así sucesivamente, lo que se necesitan son las subclaves en orden inverso, como se muestra en la figura 14. En la práctica, esto se consigue principalmente mediante el cálculo de todo el key schedule y almacenadas todas las 11, 13 o 15 subclaves, dependiendo del número de rondas que el AES está utilizando (que a su vez depende de las tres longitudes de clave soportados por AES). Este precalculo añade por lo general una pequeña latencia para la operación de descifrado en relación con el cifrado.